Permutation Based Feature Construction

Furkan Kadıoğlu

January 2019

Abstract

Missing relevant features in various machine learning and data mining tasks makes error more probable. The probability of error is degraded by constructing feature in data driven approach. We have developed a novel metric based permutation perspective in order to compare two features. Permutation based approach derives relative information between samples. This relativity gives valuable information about selected feature that differs from classical data driven approach like as cosine similarity. The distance between permutations gives fruitful information in the case of comparing two features. The main perspective of this study is to make permutation based data driven approach more clear. Permutation perspective also was used in order to generate new features by utilizing initial feature space. Relation between permutation and correlation was showed empirically.

1 Introduction

Feature construction is a process that discovers missing information about the relationships between features and augments the space of features by inferring or creating additional features [7, 6, 13]. Intuitively, combinations of original features, and not all combinations are necessary and useful. Feature construction aims to automatically transform the original representation space to a new one that can help better achieve data mining objectives improved accuracy, easy comprehensibility, truthful clusters, revealing hidden patterns, etc. Therefore, the major research issues of feature construction are the following four.[8]

How to construct new feature There are four approaches to construction of new features. They are hypothesis-driven, data-driven, knowledge-based and hybrid[6, 13]. Data Driven Approach was followed in this study because we want to use available data in order to make a solution to absence of previous information.

How to choose and design operators for feature construction Linear combination is the simplest method for new feature generation. Linear combination is used in this study because we do not want to face problem of the

complexity of operator so we can observe correlation in between distance metric and measurement of new features in an easy manner. Therefore, linear combination of features of feature spaces was selected as a operator. However, linear combination does not affect performance of the training model clearly. We wanted to clarify changes in measurement of quality in order to make results more informative. At this point, Sigmoid function was used in this study. Sigmoid function is used in neural network model as a activation function in order to obtain nonlinear relations of features.[10]. Sigmoid function was used on new feature which was attained by linear combination. New feature can be think of an array which has length is sample size. Sigmoid function was applied to all entries of new feature array.

How to use operators to construct new features efficiently There are lots of usage of possible operators. This situation force researcher to find an efficient manner at this step of researches about feature construction. Generally, researcher develop their own heuristic solution in the searching of efficient manner[8]. At this point,method of this study was decided to obtain all possible new feature from specified feature space with respect to feature dimension. After acquiring of combinations, coefficient matrix was created. All entries of coefficient matrix's are one except selected feature of the combination. Coefficient of selected feature is incremented by specified step size. Coefficient matrix's number of column depends on specified step size, specified initial point of the coefficient, specified final point of the coefficient. In the experiment of this study, step size was specified as 0.1, initial point was decided as 1.0 and final point was selected as 5.1. Therefore, column size of the coefficient matrix is 41 in this experiment.

How to measure and select useful new features There can be too many combinations. For example, dimension of initial feature space is 50. Number of binary combinations which are derived from initial feature space is 1225. And then, if we get step size, initial and final point of incrementation which are in this experiment then number of possible features will be 50,225. Therefore, in this study feature selection methods were used to attain best new features. Firstly, wrapper approach are prefered to considerate new features. In this experiment, Random Forest Classifier was selected because it is more stable with respect to neural networks. However, training a model is exhaustive when its computation cost is high. Also, result of the training success was not clear due to noise. Therefore, new measurement was needed to observe correlation in between distance metric and quality of new feature. Correlation comparison with target of the dataset provided useful information about new feature. So correlation was selected as a proxy measure. This is called as filter approach in feature selection literature [6].

2 Data Set and Feature Space

2.1 Data Set

In this study, data set which has continuous feature that is no sparse, was very useful in metrics which were defined in this study. Scikit python library is extremely useful by providing data sets and training model. Covtype data set was used in this experiment because sample size provided easy computation and are not small. Covtype data set has 54 feature. 44 of features are binary and rest of the features are continuous.

2.2 Feature Space and Feature Dimension

Reduction of initial feature is the control of sparsity for continuous features in initial feature space. Especially, sparsity control is very useful in permutation metric. Reduction of the space gives new feature space. In this experiment, unless otherwise stated eliminated feature space as called as feature space. 7 of the continuous features are not sparse in Covtype data set and Non sparse continuous features have index which are 0,1,5,6,7,8,9.

3 Distance Metrics

In this experiment, two metrics were used in order to compare two features. The first one is cosine similarity. Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them [1]. In undersampling and oversampling situation, cosine similarity is utilized in order to compare two samples which belong data set [4]. In this experiment, cosine similarity was used in order to compare two features. Every feature in training data set, can be considered as a vector. This perspective provides ability to compare two new features. This can be seen as a proxy measure that is used in feature selection when cosine similarity is discussed with this approach. Secondly, Permutation Metric that were defined as a metric by us, was used in this experiment. Actually, Cosine Similarity was considered as a benchmark for our permutation distance metrics. Similarly to cosine distance metric, Permutation metric was used in order to compare two features in feature space. Unlike to cosine similarity, feature value is not important to permutation metric. Permutation metric considers sorting with respect to features which are interested. Permutation metric composed of three step which are the following sorting with respect to selected feature, inner matrix calculation and Manhattan distance between two matrices. First of all, every sample has specified feature values. Sorting that means sorting sample with respect to these values of data set. Final of the first step gives permutation corresponding feature. There is extremely important one point which is scaling of feature. Scaling of feature gives democracy in between features which are in eliminated space. Obviously, feature scaling can be applied to continuous feature. Second operation is the calculation of inner distance matrix. Before the calculation of

inner distance matrix, describing what inner distance matrix is, will be useful in order to understand calculation and concept idea of permutation. Permutation has elements of proper range from zero to size of sample by one. Each element of inner distance matrix is the distance between two elements of permutations. For example, let's suppose there is one permutation which is 1,2,3,0,4. First row of the inner distance matrix is the following 0,3,2,1,1. First row of the inner distance matrix indicates that first sample how far from other samples. Therefore, inner distance matrix is square matrix. This matrix is symmetric. Diagonal entries of this matrix is zero. Calculation of inner distance matrix is computational expensive. However, inverse of permutation and operation of matrices in python libraries like Numpy are rescuer for this problem. This computation cost forced us to find clever way of this calculation. Elements of permutation indicates position of features. Inverse of permutation or in other words sorting of permutation gives position of every element of permutation where they are. If we know positions of entries of permutation, then we will calculate relative positions to others by using knowledge of position of specified feature. In the calculation of first row, knowledge of the location of first sample in permutation and knowledge of all elements' position in permutation can give first row of the inner distance matrix. In our examples in the above, inverse of permutation like 1,2,3,0,5 is the 3,0,1,2,4. First element of the inverse of permutation indicates that first sample is located in third order in permutation. If we subtract this element from all elements of permutation's inverse and take absolute value for all difference values, then we can attain 0,3,2,1,1. When this subtraction is applied for all sample in data set, inner distance matrix can be derived from data set. Finally, after all these calculation and sorting staff for two features, Manhattan distance or in other words Taxicab geometry is applied to two matrices. Manhattan distance is the sum of absolute value of the difference of every elements of matrices [2]. All operations which are described at the above give one real number. This real number is called as distance between two permutations.

4 Average Calculation Methods

Definitions of metrics is the first step to produce proxy measure in order to compare new features. These metrics only give relation two features. However, quality of new feature must not depend on only two features. Therefore, we need to definition of qualification of new features with respect to cosine distance metric and permutation distance metric. Unfortunately, distance value of new features and qualification of new feature how they are relate to each other, cannot be known. Therefore, new distance definition that reflects characteristic of new feature over our metrics, was needed for new features. So the new distance definition is based on calculation of distance in between new feature and other members of feature space except selected feature for production of new feature. Average of all distance to others is defined as the new distance in order to recognize new feature characteristic by using our metrics. In cosine distance, each feature in feature space set is considered as a vector and calculate

cosine values in between new features and vectors by the cosinus theorem [12]. In permutation average calculation makes important difference from cosine average distance calculation due to metric definition. Permutation distance based on permutations and this properties of permutation concept enable new distance definition records relations between samples on specified feature.

5 Correlation and Training Indicators

In previous part of the study, we defined distance metrics and characteristic of new features in order to generate new feature. However, we do not know relation qualification of generated feature and characteristic of generated feature. This relation takes important role in generation of new feature because if relation between distance and success of model is very informative with respect to optimization perspective. Measurement of qualification definition was needed in order to mine relation between metric and success regardless of relation what it is. In this study, two main approaches of feature selection literature. They are wrapper approach and filter approach. In filter approach, proxy measures are used in literature [11].Pearson correlation coefficient between two vectors is used in order to comprehend goodness of augmented feature space. Pearson's correlation coefficient is the covariance of the two variables divided by the product of their standard deviations [9]. In wrapper approach, decision of training model must be made in order to attain effects of new feature [5]. In this study, Random Forest Classifier was used as a training model. A random forest is a meta estimator that fits a number of decision tree classifiers on various subsamples of the data set and uses averaging to improve the predictive accuracy and control over-fitting [3].

6 Implementation Details and Parameters

6.1 Libraries and Why we used

Six python libraries were used in this experiment. They are Numpy, Sklearn, Pathos, Pickle, Matplot, Functools. Numpy was used in matrices' operations and recording data in tensors. Sklearn was used as data set, also it was used in training model. Random Forest Classifier belongs to Sklearn. Kernel Density Estimation was used to make more informative plots and data charts. Pathos was used in order to make computation parallel with Pool. Pickle provides serializing output of the computational costly operation. Pickle save tensors as a binary file. Therefore, repetition of computations were not needed. Matplot was used in this experiment in order to plotting result. Especially in Jupyter notebook, it is very useful for research environment. Functools provides 'partial' function which provides that functions takes parameter at different part of the program. It was impressive tools in recalling function again and again.

6.2 Data Set and Feature Processing

Covtype data set was used in this experiment. First of all, feature scaling were applied to data set by normalizing continuous features' mean and variance value. After scaling operation, train data, validation data and test data were selected randomly. Random permutation with size of samples was created and first 11,340 elements of random permutation was used as a train data. Second 11,340 elements of random permutation was used as a validation data. And then, rest of the random permutation was used as a test data 558,332. In this experiment, train data and validation data smaller than test data because distance computation was very costly not only time complexity and also memory. Like as stated previously, 'feature_space' and 'dim_feature' must be selected at the begin of experiment with respect to beforehand knowledge. 'feature_space' states selected feature which will be source for candidate new features. 'dim_feature' means that how many features will join candidate new features.

6.3 Metric Functions

Two metric functions were used in this experiment. They are 'cosine_metric' and 'permutation_metric'. 'cosine_metric' takes two arrays that have the same size with each other and returns cosine value between given vectors. 'permutation_metric' takes two permutations that have the same size with each other and returns permutation distance value for given vectors by using definition of permutation metric.

6.4 Average Calculation Functions

There are four calculation to accomplish two tasks because each task has two version which are parallel and nonparallel. Tasks are cosine based average distance calculation and permutation based average distance calculation. Parallel version of this tasks cannot be utilized in calculation of all combinations due to structure of pathos.pool. However, percentage of idle processors is approximately zero so this restriction did not cause any problem. The aim of parallel average functions are providing fast information with respect to nonparallel. When specific information will be needed, these functions should be useful. Parallel functions are 'cosine_average_parallel' and 'permutation_average_parallel'. Rest of the average functions are 'cosine_average' and 'cosine_average'. All of them take only one parameter as a new feature with size of samples. For who wonder that parallel mechanisms, there are two functions which are 'run_metric_cosine' and 'run_metric_permutation'. These functions are called in parallel functions with respect to parallel fashion. Every increment value of all combinations calls corresponding runner functions.

6.5 Measurement of Quality Functions

These functions also are called feature selection methods. They are 'correlation_comparison' and 'permutation_comparison'. Similar average calculation

functions, they take only one parameter that is new feature. However there is little difference in use of these functions with respect to size of the new features. Average calculation functions take 11,340 elements of the new feature but feature selection functions take 581,012 elements of the new feature. These comparison functions returns tuple which has test result and validation result.

6.6 Support Functions

Till at this point, definition of functions which will be used in experiment. Last two functions do not have conceptual meaning but they are very useful when functionality was considered like as debugging and minimizing of error probability. They are 'get_new_feature' and 'get_combination'.

6.6.1 Feature Generator Function

Feature generation function takes four arguments which have default values in order to handle missing parameters and attain some functionality. Four arguments follow this manner 'elements', 'coeff', 'operant' and 'random_range'. 'elements' is a an array which keeps features' index that will join generation. 'coeff' is an array like elements by considering size. It keeps features' coefficients corresponding member of elements. In default, 'operant' is selected as a numpy.sum and this is applied to corresponding columns of data set scaled features in data set. 'random_range' was used in only one case which is absence of elements and coefficents. In this case, generator function takes random two features from random range. Feature generation function returns new feature array with the size of data set 581,012.

6.6.2 Combinations Generator

Combinations generator function is 'get_combination'. This function provides all combinations of elements of 'feature_space' by taking two parameters that are 'dimension' and 'vector'. 'dimension' is the how many features will join to generation of one new candidate feature. 'vector' is the set of all features which join generation of candidate features. Combinations generator function returns two sets as a tuple. First element of this tuple is the set that contains all index combinations for feature space. Second element of this tuple is the set that contains all combinations of given vector. For example, dimension is equal to 2 and vector is 2,7. First element of tuple is the 0,0; 0,1; 1,0; 1,1. Second element is 2,2; 2,7; 7,2; 7,7. For detail information, running the function test that is located under the function. Do not forget run library cell. Input must be in this form because results must be saved as a tensor. Otherwise, we do not know which index belongs to which feature .

6.7 Experiment Parameter Selection

Some parameters must be given before computations is the selection of parameters. There are five parameters that are must be selected. These parameters can

be considered in two groups which are method parameters and increment parameters. Method parameters are 'feature_selection_method' and 'distance_method'. These method details were detailed at the above. Increment parameters are 'begin_range', 'end_range' and 'increment'. 'begin_range' is starting point for coefficient of fist element of combination. 'end_range' is ending point for coefficient of fist element of combination. 'increment' is the step size of increment.

6.8 Experiment Variable

Memory Variables Results of computations are saved as a tensor. Obviously, dimension of tensor can be change depending upon dimension of candidate features. Therefore, 'frame' is created. 'frame' is an array which has repetitive entry. All entries of array are equal to dimension of 'feature_space'. Size of this array equals to dimension of candidate features. Then the creation of 'frame', two tensors are created by using 'frame'. They are 'selection_memory' and 'distance_memory'. 'selection_memory' has dimensions which is sum of the size of 'frame' and two. Entries of 'frame' decide size of dimensions of 'selection_memory' except last two. Similarly 'selection_memory', 'distance_memory' is defined by using 'frame'. However, dimension of 'distance_memory' has a screw loss which is equal to one. This difference result from feature selection methods' returning value is to be tuple differ from distance methods. Distance methods returns one value.

Coefficient Matrix The reason of creation of this matrix is just to make computations parallel. Every column of the coefficient matrix is 'coeff' variable for runner function. Size of column can be derived by dividing difference between 'end_range' and 'begin_range' with 'increment'. Row size equals to dimension of candidate functions. In this study, all entries of coefficient matrix is one except first row. First row is equals to 'inc_range'.

Merging Operations If dimension of candidate features is equal to zero, then, this procedure can be applied because in binary combinations, when coefficient of one that is in the selected features, increases, other feature effect melts slowly. In this case, binary combinations of binary combinations are follow-up for other. In this case, new variables are needed such as 'merge_selection' and 'distance_selection'. They are very similar to other memory variable but there is only one difference. Previous memories' dimension which is specified by size of the 'inc_range' is multiplied by two due to merging operations.

7 Conclusion and Future Works

In this study, distance methods based on two metrics, feature selection methods from two different approaches and sigmoid function as an activation function were used. Observations were recorded as eight trinary combinations of distance methods and feature selection methods. Results of them were saved as

a pickle files. They are 'cosine_corr', 'cosine_train', 'permutation_corr', 'permutation_train' and their prefix versions like 'sigmo_cosine_corr'. Actually, four charts for each linear binary combinations of 'feature_space' were derived from computations. Similar to linear, four charts were derived by using activation function. Empirically, linear and nonlinear versions did not differ from other. However, usage of activation function are taking important roles, was thought. Different use of activation functions can be considered with different activation functions for future works. Training results were noisy because of small changes in success. Therefore, denoising can be studied for a future work in this context. However, different approach was prefered in this study. The behaviours of permutation distance metric and cosine distance metric were observed as an inverse proportional but contrary examples were observed, too. Relation cosine values and correlation was observed but different approaches to results are needed in order to attain clear information about it. These approaches can be studied as a future work. The clearest relation is between permutation distance and correlation in this study. If local extremum point was observed, then the neighbors of this point is the convergence for correlation. However, relation between two was not derived from experiments, exactly. Therefore, fruitful results can be extrapolated in the case of studying this relation as a future work. This kind of relations can be used in order to generate new feature as a future work.

References

[1] Cosine similarity. [Online; accessed 05-January-2019].

[2] Taxicab geometry. [Online; accessed 05-January-2019].

[3] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[4] Debashree Devi, Saroj Biswas, and Biswajit Purkayastha. Redundancy-driven modified tomek-link based undersampling: A solution to class imbalance. *Pattern Recognition Letters*, 10 2016.

[5] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273 – 324, 1997. Relevance.

[6] H Liu and H Motoda. *Feature Extraction Construction and Selection A Data Mining Perspective*. Kluwer Academic Publishers, 1998.

[7] CJ Matheus. *The Need for Constructive Induction Proc of the Eighth International Workshop on Machine Learning, pp.173-177*. 1991.

[8] H. Motoda and H. Liu. *Feature Selection Extraction and Construction*.

[9] Philip Sedgwick. Pearson's correlation coefficient. *BMJ*, 345, 2012.

[10] Sagar Sharma. *Activation Functions: Neural Networks*. 2017.

[11] Ryan J. Urbanowicz, Melissa Meeker, William La Cava, Randal S. Olson, and Jason H. Moore. Relief-based feature selection: Introduction and review. *CoRR*, abs/1711.08421, 2017.

[12] Eric W. Weisstein. "law of cosines. [Online; accessed 05-January-2019].

[13] J. Wnek and R.S. Michalski. *Hypothesis Driven Constructive Induction in A Q17-HCI A Method and Experiments, pp. 139-168.* 1994.