Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-25 (2016) 299–304

Comparing Fitness Functions for Genetic Feature Transformation

Jan Klusáček * **Václav Jirsík** **

* *Brno University of Technology The Faculty of Electrical Engineering and Communication Department of Control and Instrumentation Technická 3082/12, Královo Pole, 61600, Brno Česká republika xklusa00@stud.feec.vutbr.cz*

** *Brno University of Technology The Faculty of Electrical Engineering and Communication Department of Control and Instrumentation Technická 3082/12, Královo Pole, 61600, Brno Česká republika jirsik@feec.vutbr.cz*

Abstract: A representation of features is a very important parameter when creating machine learning models. The main goal of this paper is to introduce a way to compare feature space transformations that change this representation. It particularly deals with a comparison of a method that uses a genetic programming based on different fitness functions for transformation of feature space. The fitness function is a very important part of the genetic algorithm because it defines required properties of new feature space. Several possible fitness functions are described and compared in this paper. The process of the comparison is also introduced in this paper and selected functions are tested and their results are compared to each other. These results are compared and upsides and downsides of each method are discussed in conclusion. A part of the work is a framework used to automate processes needed to create this comparison.

Keywords: Feature space, Feature space transformation, Genetic programming, Fitness function, Machine learning

1. INTRODUCTION

One of the most used areas of a machine learning is an area of supervised learning. A goal of the supervised learning is to create a model based on given examples (consisting of both features and labels). This model should be able to assign labels to new (previously unseen) examples as accurately as possible [1]. Generally, there are two sources of errors: A variance and a bias (Friedman, 1997). The variance error is caused by noise and missing values in learning data and it could be reduced by increasing number of examples. Second source of error is bias. Bias is caused by erroneous assumptions on a character of the learned model (For example if we assume linear relation between feature and label, but it is in fact quadratic we can never fully fit model to data.). This error can be reduced by using more complex models (for example higher order polynomial) but with possibility of overfitting model [2] (Geman et al., 1992). This means that a selection of a right model is one of the most important and strait forward ways to improve the prediction accuracy. Another possibility to improve accuracy of resulting model is to preprocess input data. This preprocessing is in fact transformation of a models input space (feature space transformation).

2. FEATURE TRANSFORMATIONS

Feature space transformations cannot introduce any new information, in the best case they can preserve all information already present in original features but often they reduce an amount of information, nevertheless, they can improve resulting accuracy. This improvement is achieved by transforming feature space to a form better suited for a model. These transformations can bring other advantages apart from improving resulting accuracy. They can, for example, reduce an amount of saved data or decrease the complexity of the model. Some of these transformations will be described further.

2.1 Input cleaning

The first step of preprocessing should be cleaning of input. This process removes examples that have missing values or are another way corrupted. It's necessary to have some additional knowledge of examples to be able to decide which examples are corrupted.

2.2 Feature normalization

Raw data can be represented in different ways, using different units. This can lead to features with orders of magnitude different feature values. Models like decision trees that works with informational properties of features are generally unaffected by feature scale, but others like nearest neighbor are very sensitive to a scale because they

⋆

[1] This accuracy is measured by function called error function.

[2] Models that are more complex than necessary tend to find relations that aren't really present

10.1016/j.ifacol.2016.12.053

work with a distance between examples (Yoon and Hwang, 1995). For this reason, it's essential to normalize all features before using them. The process of normalization is a complicated problem that needs to deal among others with a problem of different distributions of features and outliers.

2.3 Feature selection

One of the most basic and the most used transformation is a feature selection. The goal of the feature selection is to remove features with a low relevancy (they are not participating on a prediction). Apart from apparent advantages of reducing an amount of stored data and complexity of the created model, it can often increase an accuracy of created model [3].

Many different feature selection algorithms are used. A large group of feature selection algorithms relies on a greedy search algorithm. They are selecting one feature after another, starting from best one. All these algorithms can be divided according to the parameter used for feature selection.

The first group of methods called wrappers uses an accuracy of created model to evaluate a quality of feature set. They start by creating a model for each available feature and evaluate their accuracy and then select the best feature. The second feature is selected same way from all possible feature subsets containing best feature and one of the unselected features. This approach is repeated until the desired number of features is acquired. These methods lead to very good accuracy if resulting subset is used with the same model. On the other hand, they have high computational requirements (repeated creation and evaluation of models) and can easily lead to overfitting.

The second group of methods is called filters. They work similarly to wrappers but they use a heuristic function [4] (criterion function) instead of the accuracy of created model. This leads to lower computational requirements and selected features are not dependent on selected model but these methods generally lead to worse accuracy than wrapper methods. All filter methods can be further divided according to their approach to the redundancy. Simple methods ignore a possibility of redundant features and therefore can select similar or even same features if they are present in the original feature space. More sophisticated methods try to address this possibility and evaluate a similarity between features. The downside of this approach is a big increase in computational requirements of these methods.

Criterion functions used to select features are very important part of feature selection process and they are used further in the experiment conducted in this paper. For this reason, few functions will be described in more details.

Information gain and information gain ratio
Information gain (Guyon and Elisseeff, 2003) is one of the most used criterion functions. This criterion is defined for nominal features and labels. Information gain is defined by equation 1 where F is the original dataset, f is a candidate for selected feature and H(F) is entropy. It is defined as a difference of entropy calculated from whole, and entropies calculated from a data set partitioned using given feature.

$$IG(F, f) = H(F) - \sum_{a \in vals(f)} H(F|a) \tag{1}$$

This criterion function naturally prefers features with more distinct values, to mitigate this problem Information gain ratio can be used. (Agrawal et al., 1993). In this case, information gain is divided by a factor that takes into account number of possible distinct values.

Area under curve
The criterion Area under the curve (AUC) is defined by (Fawcett, 2006) as the probability that the classifier based on given feature will rank a randomly chosen positive instance higher than a randomly chosen negative instance. It can be alternatively understood as a number of steps required to order all examples according to their label when they were originally ordered by feature. The main advantage of this criterion is its robustness. One problem with using AUC as criterion function is that it is defined only for binary labels. It can be however adapted for multiclass tasks using feature decomposition. There are two basic ways to decompose multiclass problem to binary. We can divide it to multiple classifications problems, each solving classification between two classes, or between one class and all other classes. In this paper, the first approach is used.

$$SU(F, f) = \frac{2 \cdot H(F, f)}{H(X) - H(Y)} \tag{2}$$

Minimum redundanct maximal relevancy
Minimum redundancy maximal relevance by (Ding and Peng, 2005) (mRMR) is a criterion that unlike others discussed so far try to address a problem of redundant information. Previously described criterions considered only relevancy between feature and a label, when selecting the best feature, but they completely ignored redundancy between selected features. This can easily lead to a selection of two identic copies (possibly with some noise) if they are present in the original feature set. This is undesirable because redundant features don't provide any additional information and undermine the whole purpose of the feature selection. Its main disadvantage is higher computational requirements that rise quadratically with a number of selected features (because redundancy with each previously selected feature has to be calculated for every candidate feature).

2.4 Principal component analysis

Principal component analysis (PCA) described by (Jolliffe, 2002) is method used to transform (and possibly reduce dimensionality) feature space. This method works with complete feature space instead of one feature after another in contrast to filter and wrapper methods. PCA is searching for projections from the original feature space to a new one. It is searching for the new features as a

[3] Models like decision trees are mostly immune to abundant features. Models like nearest neighbor are very sensitive on the dimensionality of feature space and can greatly profit from its reduction

[4] There are many available functions like information Gain, Relief, Area under curve