Chapter 16
A Modified Stacking Ensemble Machine Learning Algorithm Using Genetic Algorithms

Riyaz Sikora
University of Texas – Arlington, USA

O'la Al-Laymoun
University of Texas – Arlington, USA

ABSTRACT

Distributed data mining and ensemble learning are two methods that aim to address the issue of data scaling, which is required to process the large amount of data collected these days. Distributed data mining looks at how data that is distributed can be effectively mined without having to collect the data at one central location. Ensemble learning techniques aim to create a meta-classifier by combining several classifiers created on the same data and improve their performance. In this chapter, the authors use concepts from both of these fields to create a modified and improved version of the standard stacking ensemble learning technique by using a Genetic Algorithm (GA) for creating the meta-classifier. They test the GA-based stacking algorithm on ten data sets from the UCI Data Repository and show the improvement in performance over the individual learning algorithms as well as over the standard stacking algorithm.

1. INTRODUCTION

According to some estimates we create 2.5 quintillion bytes of data every day, with 90% of the data in the world today being created in the last two years alone (IBM, 2012). This massive increase in the data being collected is a result of ubiquitous information gathering devices, such as sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals to name a few. With the increased need for doing data mining and analyses on this big data, there is a need for scaling up and improving the performance of traditional data mining and learning algorithms. Two related fields of distributed data mining and ensemble learning aim to ad-

DOI: 10.4018/978-1-5225-1759-7.ch016

dress this scaling issue. Distributed data mining looks at how data that is distributed can be effectively mined without having to collect the data at one central location (Zeng et. al., 2012). Ensemble learning techniques aim to create a meta-classifier by combining several classifiers, typically by voting, created on the same data and improve their performance (Dzeroski & Zenko, 2004; Optiz & Maclin, 1999). Ensembles are usually used to overcome three types of problems associated with base learning algorithms: the statistical problem; the computational problem; and the representational problem (Dietterich, 2002). When the sample size of a data set is too small in comparison with the possible space of hypotheses, a learning algorithm might choose to output a hypothesis from a set of hypotheses having the same accuracy on the training data. The statistical problem arises in such cases if the chosen hypothesis cannot predict new data. The computational problem occurs when a learning algorithm gets stuck in a wrong local minimum instead of finding the best hypothesis within the hypotheses space. Finally, the representational problem happens when no hypothesis within the hypotheses space is a good approximation to the true function *f*. In general, ensembles have been found to be more accurate than any of their single component classifiers (Optiz & Maclin, 1999; Pal, 2007).

The extant literature on machine learning proposes many approaches regarding designing ensembles. One approach is to create an ensemble by manipulating the training data, the input features, or the output labels of the training data, or by injecting randomness into the learning algorithm (Dietterich, 2002). For example, Bagging learning ensembles, or bootstrap aggregating, introduced by Breiman (1996), generates multiple training datasets with the same sample size as the original dataset using random sampling with replacement. A learning algorithm is then applied on each of the bootstrap samples and the resulting classifiers are aggregated using a plurality vote when predicting a class and using averaging of the prediction of the different classifiers when predicting a numeric value. While Bagging can significantly improve the performance of unstable learning algorithms such as neural networks, it can be ineffective or even slightly deteriorate the performance of the stable ones such as k- nearest neighbor methods (Breiman, 1996).

An alternative approach is to create a generalized additive model which chooses the weighted sum of the component models that best fit the training data. For example, Boosting methods can be used to improve the accuracy of any "weak" learning algorithm by assigning higher weights for the misclassified instances. The same algorithm is then reapplied several times and weighted voting is used to combine the predictions of the resulting series of classifiers (Pal, 2007). Examples of Boosting methods include AdaBoost, AdaBoost.M1 and AdaBoost.M2 which were proposed by Freund and Schapire (1996). In a study conducted by Dietterich (2000) comparing the performance of the three ensemble methods Bagging, Randomizing and Boosting using C4.5 on 33 datasets with little or no noise, AdaBoost produced the best results. When classification noise was added to the data sets, Bagging provided superior performance to AdaBoost and Randomized C4.5 through increasing the diversity of the generated classifiers. Another approach is to apply different learning algorithms to a single dataset. Then the predictions of the different classifiers are combined and used by a meta-level-classifier to generate a final hypothesis. This technique is called "stacking" (Dzeroski & Zenko, 2004).

In this paper we use concepts from ensemble learning and distributed data mining to create a modified and improved version of the stacking learning technique by using a genetic algorithm (GA) for creating the meta-classifier. We use Weka-3, the suite of machine learning and data mining algorithms written in Java for all our experiments. We use concepts from distributed data mining to study different ways of distributing the data and use the concept of stacking ensemble learning to use different learning algorithms on each sub-set and create a meta-classifier using a genetic algorithm. We test the GA-based stacking