Engineering Applications of Artificial Intelligence ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Bayesian posterior misclassification error risk distributions for ensemble classifiers

Parag C. Pendharkar *

School of Business Administration, Pennsylvania State University at Harrisburg, 777 West Harrisburg Pike, Middletown, PA 17057, United States

ARTICLE INFO

Article history:
Received 13 February 2016
Received in revised form
13 August 2016
Accepted 1 September 2016

Keywords:
Misclassification cost risk
Ensembles
Classification

ABSTRACT

Computing risk-based misclassification error density distribution for ensembles is an important yet difficult task. Bayesian methods provide one way to estimate these density distributions. In this paper, Bayesian modeling approach is used to compute posterior misclassification error density distributions for both binary and non-binary classifiers. Real-world datasets and holdout samples are used to illustrate computation of posterior misclassification error distributions. These posterior error distributions are very useful to compare ensembles, and provide risk-based misclassification cost estimates.

1. Introduction

There is a recent surge in interest in applying risk management approaches for management science (Wu, 2016) and data mining area (Wu and Birge, 2016). A range of applications have appeared from enterprise risk management (Wu et al., 2015), cost accounting (Wu et al., 2014), dynamic pricing (Wu and Wu, 2016), merger evaluation (Wu et al., 2014) and data mining (Wu et al., 2014).

Classification problem is one of the most studied problem in data mining. It is well known that classification methods are often unstable and sensitive to noise (Du et al., 2015) in training datasets (Bhattacharyya and Pendharkar, 1998), and they need to incorporate risk management techniques. One way to manage risk in these methods is to improve stability of these methods by combining multiple predictions of multiple classifiers into a single prediction (Breiman, 1996). Such techniques are known as ensembles in data mining literature (Tan et al., 2006). The Bias-Variance framework provides a formal theory in analyzing the behavior of ensembles (Tan et al., 2006). The bias component in the Bias-Variance framework deals with the assumption related to a classifier about the nature of its classification function decision boundary. The variance component deals with the differences in composition of training dataset that may lead to different classification function decision boundaries. Averaging classifier decisions results in lower error risk between the unknown true decision boundary and known ensemble decision boundary (Mendes-Moreira et al., 2012).

There are different ways an ensemble can be created. Most approaches vary two parameters: sampling with training examples and voting mechanism (i.e., weighting) of individual classifiers in an ensemble. A simple ensemble (called Bagging) simply aggregates class forecast from its classifiers that are either training on one single training dataset or using repeated samples from a common training dataset (bootstrap sampling). For small datasets, bootstrap sampling is often necessary for better generalization (Witten et al., 2011). Other approaches such as Boosting approach deals with weighting schemes for bootstrap sampling that assign higher probability to aid higher selection of examples that are difficult to classify. Bayesian modeling approaches assign higher weights to a classifier that has higher predictive performance (Lindstrom et al., 2015).

While the Bias-Variance framework recognizes ensemble errors, no study in literature appears to have focused on learning probability density distributions of such errors. Availability of such probability density distributions will allow decision-makers to select, design and compare different ensemble methods to determine the suitability of different methods for a problem at hand. Since data and problem domain play such an important part of ensemble performance, the error densities must be conditioned on examples from the problem domain. The Bayesian data analysis framework provides excellent tools for learning such posterior error distributions (Zaidan et al., 2015).

In this research, a Bayesian framework for estimating posterior classification error and correct classification distributions is proposed. While a reader may note that classification approaches require tweaking of several parameters to improve their

* Corresponding author.
E-mail address: pxp19@psu.edu
URL: http://www.personal.psu.edu/pxp19/

http://dx.doi.org/10.1016/j.engappai.2016.09.001
0952-1976/

performance, and ensemble techniques require consideration of several design and training data sampling issues, minimal treatment to such issues is provided in this research to keep focus only on learning and application of Bayesian posterior distribution methods. A simple Bagging ensemble is considered where classification techniques are trained on one common dataset and each classification technique has equal weight (voting right) for final prediction. Some experimentation on parameters related to classification techniques is performed before settling on the final set of parameters. The primary focus of this research is learning posterior distributions given that the votes of individual classifiers from an ensemble are available. Sample code from simulations (in Appendix A) used in this study is provided to aid other researchers in using the proposed Bayesian data analysis methods. The rest of the paper is organized as follows: In Section 2 the Bayesian model for learning posterior error and correct classification posterior distributions is proposed. In Section 3, different classification techniques used in Bagging ensemble are described. In Section 4, description of data used in this study is provided, which is followed by experiments and results. In Section 5, the paper concludes with a summary and directions for future research.

2. The Bayesian model for posterior distributions in ensembles

To introduce Bayesian models for ensembles, a binary classification problem is considered. The extension of current description to multiclass problems is straight forward. Assume a test dataset containing $n>1$ examples and $k>1$ classifiers in an ensemble. The primary objective in this section is to learn posterior distribution probabilities for the ensemble. Let these probabilities be represented by a vector $\varphi=[\varphi^{11}, \varphi^{12}, \varphi^{21}, \varphi^{22}]^T$, where φ^{11} is defined as the probability that an ensemble will classify an example in class 1 when it actually belongs to class 1, and φ^{12} is the probability that an ensemble will classify an example in class 1 when it actually belongs to class 2. The remaining components are defined similarly. Also, the probabilities are normalized and sum of all components adds exactly to 1. A traditional data mining setup is considered, where the actual class assignments for both training and test datasets are known.

Bayesian models require additional consideration of example independence, where individual examples in test dataset are independent of each other, and define a similar probability vector $\theta_i = \left[\theta_i^{11}, \theta_i^{12}, \theta_i^{21}, \theta_i^{22}\right]^T$, which represents ensemble classification probabilities for the ith ($i=1..,n$) example. Additionally, a vote vector $y_i = \left[y_i^1, y_i^2\right]^T$ is considered, where y_i^1 is an integer representing number of classifiers in k ensemble predicting the test example i belonging to class 1 and y_i^2 is an integer representing the number of classifiers predicting the test example i belonging to class 2. Thus, for k-classifier ensemble $y_i^1 + y_i^2=k$. The vector y_i is the ensemble test data set prediction for example i and represents available data upon which posterior distributions for θ_i are conditioned. The collection of all such predictions for entire dataset is a class-by-test examples matrix $Y=[y_1, \ldots, y_n]$. Focusing on the components of θ_i these posterior distributions may be represents as: $p(\theta_i^{11}|\, y_i)$, $p(\theta_i^{12}|\, y_i)$, $p(\theta_i^{21}|\, y_i)$, and $p(\theta_i^{22}|\, y_i)$. Assuming that the components of θ_i are independent and identically distributed, a relationship between components of vector θ_i and vector φ can be established. This relationship between posterior distributions for components φ^{11} and θ_i^{11} may be established using following expression:

$$p\left(\varphi^{11} \mid Y\right) = \prod_{i=1}^{n} p\left(\theta_i^{11} \mid y_i\right). \tag{2.1}$$

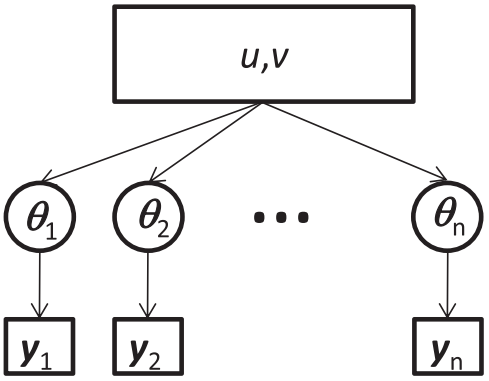

Fig. 1. Structure for traditional Bayesian Model.

Using the Bayesian theorem, the right hand side component may be written as follows:

$$p\left(\theta_i^{11} \mid y_i\right) \propto p\left(y_i \mid \theta_i^{11}\right) \times p\left(\theta_i^{11}\right). \tag{2.2}$$

The posterior distributions for other components can be similarly defined and in a vector form as follows:

$$p\left(\theta_i \mid y_i\right) \propto p\left(y_i \mid \theta_i\right) \times p(\theta_i). \tag{2.3}$$

The structure of traditional Bayesian model implementing right hand side of Eq. (2.3) is shown in Fig. 1. When classification problem is binary, conjugate Binomial likelihood and Beta prior are used in this research. Mathematically, the likelihood and prior distribution for binary classification problems are represented as follows:

$$y_i \sim Binomial(\theta_i, k), \text{and} \tag{2.4}$$

$$\theta_i \sim Beta(u, v) \tag{2.5}$$

Posterior distributions for $p(\varphi|Y)$ are computed using Bayesian Markov Chain Monte Carlo (MCMC) simulations using the Winbugs software. Low information prior with values of $u=1$ and $v=1$ are used in the simulations. For classification problems with more than two classes, following likelihood and prior are used:

$$y_i \sim Multinomial(\theta_i, k), \text{ and} \tag{2.6}$$

$$\theta_i \sim Dirichlet(e). \tag{2.7}$$

In Eq. (2.7), e is a vector of ones with its dimension equal to number of classes (or groups say $g \geq 2$) in the classification problem.

3. Classification techniques used in ensembles

For describing the classification algorithms used in this research, a classification problem is assumed to be consisting of N training examples, where each example is denoted by a tuple (x_i, z_i) ($i=1,\ldots,N$). The vector $x_i=[x_{i1},\ldots,x_{id}]^T$ corresponds to decision-making attribute set for the ith example, and the variable z_i is an integer denoting its class label. The individual techniques used in this research are described in following sub-sections.

3.1. Support vector machines

Support vector machines (SVM) were introduced by Vapnik (1995), and assume that the class labels for a binary classification problem are given by $z_i \in \{-1,1\}$. The SVM learns linear