Knowledge-Based Systems 90 (2015) 224–238

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Gravitational fixed radius nearest neighbor for imbalanced problem

Yujin Zhu, Zhe Wang*, Daqi Gao*

Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, PR China

ARTICLE INFO

Article history:
Received 18 March 2015
Revised 20 August 2015
Accepted 14 September 2015
Available online 3 October 2015

Keywords:
Fixed radius search
Nearest neighbor rule
Imbalanced data
Pattern classification

ABSTRACT

This paper proposes a novel learning model that introduces the calculation of the pairwise gravitation of the selected patterns into the classical fixed radius nearest neighbor method, in order to overcome the drawback of the original nearest neighbor rule when dealing with imbalanced data. The traditional k nearest neighbor rule is considered to lose power on imbalanced datasets because the final decision might be dominated by the patterns from negative classes in spite of the distance measurements. Differently from the existing modified nearest neighbor learning model, the proposed method named GFRNN has a simple structure and thus becomes easy to work. Moreover, all parameters of GFRNN do not need initializing or coordinating during the whole learning procedure. In practice, GFRNN first selects patterns as *candidates* out of the training set under the fixed radius nearest neighbor rule, and then introduces the metric based on the modified law of gravitation in the physical world to measure the distance between the query pattern and each *candidate*. Finally, GFRNN makes the decision based on the sum of all the corresponding gravitational forces from the *candidates* on the query pattern. The experimental comparison validates both the effectiveness and the efficiency of GFRNN on forty imbalanced datasets, comparing to nine typical methods. As a conclusion, the contribution of this paper is constructing a new simple nearest neighbor architecture to deal with imbalanced classification effectively without any manually parameter coordination, and further expanding the family of the nearest neighbor based rules.

1. Introduction

Class distribution is defined as the proportion of patterns belonging to different classes in a dataset and plays a pivotal role in pattern recognition [11,12,38]. As a special case of class distribution, the imbalanced dataset is the case that the number of patterns from one class is far less than those belonging to the other classes [3,5,12,15]. Further in real-world classification tasks, the class with less patterns generally attracts more interests than the others and then is defined as the positive class [2,11,24]. Correspondingly, the other classes with more patterns are defined as the negative classes. To be convenient, this paper abbreviates the positive class (i.e., the minority class) to POS and the negative classes (i.e., the majority classes) to NEG. Generally in binary-class imbalanced problems, one indicator named the Imbalance Ratio (*IR*) [26] is defined in Eq. (1) to measure the imbalance degree of one dataset:

$$IR = \frac{N_{NEG}}{N_{POS}}, \tag{1}$$

where N_{NEG} and N_{POS} mean the number of patterns from NEG and POS, respectively.

The previous research studies [2,3,11,15] have revealed that the traditional classification methods deteriorate more or less when dealing with the imbalanced datasets. The classical k Nearest Neighbor search algorithm (kNN) has no exception [19,20,23,33]. kNN used to be evaluated as one of the top 10 algorithms in data mining [37] because of its simple but powerful principle, which recognizes a query pattern only based on the most frequent class distribution of its k nearest neighbors in testing steps [8]. However, kNN might be misled in imbalanced problems because that 1) the decision of kNN might be dominated by the NEG patterns around the query pattern [23] and 2) the selection of k is data-dependent and difficult to tune [33]. For instance, a kNN with $k = 7$ is prone to classify a query pattern into NEG in a binary-case, in spite of the fact that the two nearest neighbors belong to POS and the other five farside patterns belong to NEG.

The existing solutions for imbalanced problems can be categorized into three parts: firstly, the data-oriented methods use sampling techniques to achieve the equilibrium of the class distribution, such as the typical over-sampling method named SMOTE [7] that increases the size of POS with the synthetic patterns. Secondly, the cost-sensitive methods consider the penalties associated with misclassifying patterns [34]. Finally, the ensemble methods improve the performance of each used classifier [11]. Nevertheless, research

* Corresponding authors. Tel.: +86 15000779526.
E-mail addresses: wangzhe@ecust.edu.cn, wangzhe_hy@nuaa.edu.cn (Z. Wang), gaodaqi@ecust.edu.cn (D. Gao).

http://dx.doi.org/10.1016/j.knosys.2015.09.015
0950-7051/

studies on *k*NN with imbalanced datasets are far from enough [20]. In general, the corresponding interests of *k*NN for imbalanced problem can be divided into two branches: the pattern-oriented methods aiming to amplify the effect of each POS pattern [20,39] and the distribution-oriented methods trying to acquire more informative prior knowledge of global distribution [10,19,23]. For the first branch, one typical pattern-oriented method is the *k* Exemplar-based Nearest Neighbor (ENN) [20] that first selects the pivot POS patterns and then expands the boundary of them into Gaussian balls. In detail, ENN uses a newly-defined distance between the query pattern and the surface of the ball of one pivot POS pattern instead of the original distance between them, leading to a nearer connection between the query one and the pivot one. Afterwards, the Positive-biased Nearest Neighbor (PNN) [39] is proposed to boost ENN by dynamically comparing the distance between the *k*th nearest local neighbor and the query to the distance between the *r*th nearest POS pattern and the query. According to the rule of PNN, one of the two parameters *k* and *r* is finally selected to balance the local distribution of the binary-class patterns. That is, the search area of the query pattern is dependent on the value of the two parameters. Differently from ENN, PNN has no training steps [39]. As a result, PNN can deal with test patterns faster. Another idea is to learn the relationship between the query and its neighbors to correct the substantial bias to major class iteratively [13]. In detail, the coordination of each iteration considers the infection from patterns of both intra-classes and inter-classes as the weight and adopts the Geometric Means metric (GM) as the measurement [13].

As for the second branch of *k*NN-related methods for imbalanced problems, the typical distribution-oriented strategies are listed as follows. First, the Class Confidence Weighted *k*NN algorithm (CCW-*k*NN) [23] obtains the different weights by calculating the mixture models or Bayesian networks and then imposes the weights on various neighbors as the confidence. Besides, the Class Based Weighted *k* Nearest Neighbor [10] generates weights for each pattern by calculating the rate of misclassification of each class by the original *k*NN. In addition, the Class Conditional Nearest Neighbor Distribution (CCNND) [19] compares the query pattern to patterns of each class in turn, and thus finds the most eligible class, in which the query pattern defeats the most intra-class patterns in terms of the pointwise comparison of distances to their *k* neighbors. Moreover, the Informative *k* Nearest Neighbor (I*k*NN) containing the localized version (LI-*k*NN) and the globalized version (GI-*k*NN) introduces a new metric that measures the informativeness between patterns first, and then finds the top *I* nearest patterns as the final *candidate*s from the basic *k* nearest neighbors [33]. Finally, there are methods combining both of the pattern and the distribution-oriented idea, e.g., the Fuzzy-Rough *k* Nearest Neighbor Algorithm [14] constructs relation between the query patterns and its neighbors based on the fuzzy membership function, while some related approaches [21,22] are proposed based on the fuzzy knowledge.

It can be concluded that most of the mentioned nearest neighbor algorithms, to a certain extent, intend to learn and utilize global information to make a progress, but there is still possibility for overcoming the existing drawbacks and improving the previous work: firstly, the learning models usually seem too complex and not easy to be approached without adopting special data structures; secondly, some of the classifiers have to tune many parameters and the process for finding optimal parameters costs extra time; thirdly, the influence of the global information might be weakened during the training process. In this paper, we intend to find a simpler but more robust way to make a progress. Differently from the existing methods, we first consider to separate rather than combine the processes to acquire both global and local knowledge. In detail, we first try to eliminate ineligible patterns globally and then deal with the surviving ones that are called as *candidate*s in this paper. To fulfill the global search task, we prefer to adopt the Fixed Radius Nearest Neighbor search strategy (FRNN) [4,25] instead of the traditional *k*NN. Further inspired and modified by the typical gravitation-based methods including the Data Gravitation based Classification (GDC) [27,28] strategy and the Gravitational Search Algorithm (GSA) [29,30], we aim to design and introduce a new local decision criterion based on the gravitational rule into the FRNN. Furthermore, all steps are not expected to require any parameter input manually. To our best knowledge, it is the first time to propose the gravitation-inspired FRNN without any manually-coordinated parameters. To be convenient, the novel method is called GFRNN in short.

The major contribution of this paper lies in the following aspects:

- Motivation: This paper tries to propose a new easy-to-approach nearest neighbor learning model by introducing the calculation of universal gravitation into the traditional FRNN, in order to overcome the drawback of the original nearest neighbor rule on imbalanced classification problems.
- Novelty: The proposed method is expected to eliminate unnecessary patterns through the FRNN strategy and makes the final decision according to the sum of the gravitational forces between the query and the surviving patterns. Moreover, none of parameters are manually set or coordinated in the whole learning procedure.
- Experiments: The experiments are designed to compare GFRNN to some typical algorithms. Specially, nine classifiers including CCNND [19], ENN [20], LI-*k*NN [33], PNN [39], *k*NN [8], FR*k*NN [25], C4.5 [6], Logistic Regression (LR) [16], and the Support Vector Machine (SVM) [35] are considered. Finally, the results demonstrate the effectiveness and the efficiency of the proposed method.

The rest of this paper is organized as follows. Section 2 presents the architecture of the proposed method and demonstrates relevant analyses on it. Section 3 reports on all the experimental results. Finally, conclusions are given in Section 4.

2. Description of GFRNN

In this paper, we focus on the binary-class recognition for imbalanced datasets even though GFRNN can be generalized into the multi-class problems. At first, we suppose that there is an binary-class imbalanced dataset. The training set of the dataset is X_{All} that includes the set of POS: $X_{POS} = \{(\boldsymbol{x}_1, \varphi_1), (\boldsymbol{x}_2, \varphi_1), \ldots, (\boldsymbol{x}_{n_{POS}}, \varphi_1)\}$, and the set of NEG: $X_{NEG} = \{(\boldsymbol{x}_{n_{POS}+1}, \varphi_2), (\boldsymbol{x}_{n_{POS}+2}, \varphi_2), \ldots, (\boldsymbol{x}_{n_{POS}+n_{NEG}}, \varphi_2)\}$, where $\varphi_i \in \{1, -1\}$ is the label, $i = 1$ for POS and $i = 2$ for NEG. n_{POS} and n_{NEG} are the number of training patterns belonging to POS and NEG, respectively. Therefore, the number of the total training patterns can be written as:

$$n_{ALL} = n_{POS} + n_{NEG}. \tag{2}$$

Moreover, we define the query pattern $\boldsymbol{y}$ and formulate the function $\mathrm{d}(\cdot)$ in Eq. (3) to measure the distance between two patterns. To be simple, we adopt Euclidean distance here though any appropriate measurements could be used.

$$\mathrm{d}(\boldsymbol{x}_p, \boldsymbol{x}_q) = ||\boldsymbol{x}_p - \boldsymbol{x}_q||_2. \tag{3}$$

In addition, the training pattern that survives from the FRNN is named *candidate* in sequent parts. Finally, it can be summarized that the formulation of GFRNN includes three main steps: first selecting *candidate*s around $\boldsymbol{y}$ through FRNN, then calculating the distance between $\boldsymbol{y}$ and each *candidate* based on the simplified law of gravitation, and making the decision according to the sum of the gravitational forces of all $\boldsymbol{y}$-*candidate* pairs at last.

2.1. Candidate*s selection based on the Fixed Radius Nearest Neighbor search (FRNN)*

According to the literature [25], there are three most popular nearest neighbor search methods: *k*NN, FRNN, and the combination of *k*NN and FRNN (i.e., FR*k*NN). At first, *k*NN can be defined as [25]:

$$k\mathrm{NN}(\boldsymbol{y}, X_{All}, k) = A, \tag{4}$$