KRNN: k Rare-class Nearest Neighbour Classification

Xiuzhen Zhang[a,*], Yuxuan Li[b], Ramamohanarao Kotagiri[b], Lifang Wu[c], Zahir Tari[a], Mohamed Cheriet[d]

[a] *RMIT University, Australia*
[b] *The University of Melbourne, Australia*
[c] *Beijing University of Technology, PR China*
[d] *The University of Quebec (ETS), Canada*

Abstract

Imbalanced classification is a challenging problem. Re-sampling and cost-sensitive learning are *global* strategies for generality-oriented algorithms such as the decision tree, targeting inter-class imbalance. We research *local* strategies for the specificity-oriented learning algorithms like the k Nearest Neighbour (KNN) to address the within-class imbalance issue of positive data sparsity. We propose an algorithm k Rare-class Nearest Neighbour, or KRNN, by directly adjusting the induction bias of KNN. We propose to form dynamic query neighbourhoods, and to further adjust the positive posterior probability estimation to bias classification towards the rare class. We conducted extensive experiments on thirty real-world and artificial datasets to evaluate the performance of KRNN. Our experiments showed that KRNN significantly improved KNN for classification of the rare class, and often outperformed re-sampling and cost-sensitive learning strategies with generality-oriented base learners.

Keywords: Imbalanced classification, nearest neighbour classification, KNN, re-sampling, cost-sensitive learning

*Corresponding author. School of Computer Science and IT, RMIT University, GPO Box 2476, Melbourne, VIC 3001, Australia.
Email address: xiuzhen.zhang@rmit.edu.au (Xiuzhen Zhang)

1. Introduction

In many real-world applications class distribution is highly imbalanced. Typically the objective of imbalanced classification is to achieve accurate classification for each class, especially the rare, or positive class [1, 2]. (Hereafter rare class and positive class are used interchangeably.) For example, identifying bugs in source code is of uttermost importance in software development projects, but typically bugs only occur at a rate of 5-10% [3]. Other examples for skewed class distribution include oil spills in satellite radar images [1] and fraudulent calls [4]. Imbalanced classification has been recognised as a challenging and long-standing problem in data mining research [2, 5].

Existing imbalanced learning strategies can be broadly grouped into three categories, re-sampling, cost-sensitive learning, and learning algorithm-specific approaches. Re-sampling is a generic strategy for any base learning algorithm. It generally over-samples the minority class and/or under-samples the majority class to re-balance the class distribution [6, 7]. Cost-sensitive learning is also a generic strategy that generally associates higher misclassification cost with the rare class so that classification decisions are biased towards the rare class [8, 9, 10]. Re-sampling and cost-sensitive learning involve extra training and often incur extra computation cost. Directly adjusting the induction bias of specific learning models is an algorithmic strategy [11, 12, 13, 14, 15].

There are generality-oriented and specificity-oriented learning algorithms [2]. For generality-oriented learners, abstract classification models are inducted (trained) from the training data. The decision tree and the support vector machine are examples of generality-oriented learning schemes. For specificity-oriented learners no explicit models are inferred from training data, and direct classification decision is reached locally for given data points or their vicinity in the training instance space [16]. The k Nearest Neighbour (KNN) and instance-based learning algorithms [17] are representatives for the specificity-oriented scheme. KNN is a computationally-simple specificity-oriented algorithm.

In addition to inter-class imbalance, imbalanced datasets also have the within-

class imbalance issue of positive data sparsity [18, 2]. Existing studies have mostly focused on resampling and cost-sensitive learning strategies for generality-oriented learning algorithms like the decision tree [6, 7, 8, 9, 10], but they are ineffective for specificity-oriented algorithms like KNN, as shown in our experiments (Setion 7.2). Such results may be explained by the global nature of resampling and cost-sensitive learning, which only target inter-class imbalance. However, these strategies do not address the issue of inaccurate probability estimation due to positive data sparsity [19]. Although re-sampling ensures overall equal prior class probabilities, it can not improve posterior class probability estimation for instances when positive instances are scarce in local regions. Similarly, with cost-sensitive learning, posterior class probability estimation is also important for computing the expected minimal cost, but the issue of positive data sparsity in local regions is largely ignored.

To address the within-class imbalance issue of positive data sparsity, we propose local strategies to directly adjust the induction bias of specificity-oriented learning algorithms. We choose KNN as a representative specificity-oriented learning algorithm due to its simplicity and low learning cost. We propose the k Rare-class Nearest Neighbour (KRNN) classification algorithm, where dynamic local query neighbourhoods are formed that contain at least k positive nearest neighbours and the positive posterior probability estimation is biased towards the rare class based on the size and positive distribution in local regions. To the best of our knowledge, our approach of positive-biased posterior probability estimation based on local positive distributions is the first of its kind.

We conducted extensive experiments on 14 real-world and 16 artificial imbalanced datasets to benchmark KRNN against state-of-the-art imbalanced learning strategies. KRNN was benchmarked against the widely used SMOTE [7] re-sampling (synthetic oversampling for the rare class) and MetaCost [8] cost-sensitive learning strategies with popular generality-oriented learners the decision tree [20] and the support vector machine [21]. KRNN was also benchmarked against SMOTE and MetaCost with the specificity-oriented learner KNN and against recent KNN-based algorithmic approaches, namely Positive-biased Near-

est Neighbour (PNN) [15], which adjusts the positive probability estimation, Exemplar-based Nearest Neighbour (ENN) [13][1], which inducts generalised concepts, and Class-Weighted k Nearest Neighbour (CCW-KNN) [14], which computes class weights [14] for each classification decision.

We evaluated the imbalanced classification performance using the receiver operating characteristic (ROC) curve, which characterises classification performance at varying misclassification costs, including Area Under ROC curve (AUC) [23] and ROC Convex Hull analysis [24]. Our experiments showed that, in terms of AUC, KRNN consistently outperformed the KNN family of algorithms, including SMOTE-KNN and MetaCost-KNN. KRNN also consistently achieved AUC significantly higher than SMOTE and MetaCost with the decision tree. KRNN often achieved the optimal classification result on the Convex Hull, especially at low misclassification costs.

We make two main contributions in this paper:

- In contrast to the popular re-sampling and cost-sensitive learning strategies, which are global strategies targeting inter-class imbalance, we propose local strategies to directly adjust the induction bias of KNN to combat the within-class imbalance of positive data sparsity.

- We propose an intuitive, simple approach to positive-biased posterior class probability estimation based on the positive distribution in local regions, which does not quire extra training.

2. Related Work

In the literature strategies for imbalanced classification mainly fall under three categories: re-sampling [6, 7], cost-sensitive learning [25, 8, 19, 10, 26], and directly adjusting the induction bias of learning algorithms [11, 12, 27, 13, 14, 15]. The strategies can also be combined [28]. Common re-sampling

[1] ENN is different from the Edited Nearest Neighbour algorithm [22].

techniques include random over-sampling and under-sampling, as well as intelligent re-sampling. Chawla et al. [7] proposed Synthetic Minority Over-sampling TEchnique (SMOTE) to over-sample the rare class by creating artificial but non-replicated rare-class samples. Alhammady [29] proposed to over-sample the rare class by creating artificial rare-class samples based on patterns characterising differences among classes. In [30] an adaptive synthetic sampling approach was proposed to generate more rare-class samples that are harder to learn.

Typically the cost-sensitive learning strategy associates higher cost for incorrectly classifying instances from the rare class, although the specific cost information is not always available. Domingos [8] proposed a generic re-costing method called MetaCost. Experiments show that MetaCost reduces costs compared to the cost-blind classifier, using C4.5Rules (a decision tree classifier) as the base learner. Sun et al. [26] proposed to use boosting techniques for imbalanced learning, and introduced three cost-sensitive boosting algorithms. Based on the observation that boosting tends to overweight the minority class instances, in [10] cost-sensitive learning is combined with feature selection for effective imbalance learning. It should be noted that the re-sampling and cost-sensitive learning strategies discussed above all use generality-oriented learning algorithms like the decision tree [20] and the support vector machine (SVM) [21] as the base learning algorithm.

There have been comparative studies on the effectiveness of re-sampling and cost-sensitive learning for imbalanced classification [19, 31]. In [19] Elkan studied the problem of optimal learning with different misclassification costs and showed that compared with adjusting cost for each class, re-sampling has little effect on the decision tree and Bayesian learning algorithms. In [31] Weiss et al. raised the question "why doesn't the cost-sensitive learning algorithm perform better given the known drawbacks with sampling?" In [28], SMOTE sampling and cost-sensitive learning are combined, using the SVM base learner. The approach is shown to outperform any separate approach.

Directly adjusting the induction bias of specific learners is a third strategy for imbalanced classification. One class of studies focus on how to make the

induction bias of generality-oriented classification algorithms like the decision tree more specific so as to improve their performance for the rare class. Holte et al. [11] modified the bias of the decision tree classifier CN2 by using the maximum generality bias for large disjuncts and a selective specificity bias for small disjuncts, where a disjunct is a sub-cluster of samples corresponding to a sub-concept for a class. In another piece of work [12], a hybrid approach is adopted to address the imbalanced problem – the C4.5 decision tree classifier [20] is used as the base learner, and an instance-based classifier is used if small disjuncts are encountered. Similar approaches were introduced [32, 33], combining a genetic algorithm and the C4.5 decision tree. In [34] kernel-based classifiers are proposed to optimise model generalisation for imbalance classification.

Another class of studies for directly adjusting the induction bias of learners focus on adjusting the induction bias of specificity algorithms, specifically KNN [27, 13, 14, 15]. In both [13] and [14] a training stage is introduced to identify exemplar positives or to derive weights for "signifiant" training instances. Li and Zhang [13] proposed Exemplar-based Nearest Neighbour (ENN), where exemplar positives are those that can be safely generalised to a positive region. Liu and Chawla [14] proposed Class Confidence-Weighted k Nearest Neighbour (CCW-KNN), where class confidence (probability) for an attribute-value pair is estimated. Dubey and Pudi [27] proposed weighted KNN by estimating the class weight for each instance in the training instance space. In all these three pieces of work an extra training stage is introduced where extra computation is needed for learning. In contrast, Positive-biased Nearest Neighbour (PNN) [15] is a simple approach directly adjusting the class probability estimation from the local neighbourhood for query instances, without extra computation cost for learning from training instances. The class probability estimation of PNN however, is somewhat ad hoc, where for a query neighbourhood where positives are underrepresented, the positive probability is crudely estimated as $\lceil \frac{k}{2} \rceil / k$.

Notably there have been studies [35] comparing the performance of specificity-oriented and generality-oriented algorithms for imbalance learning. The performance of KNN for class imbalance is examined in comparison to several

generality-oriented learning algorithms including C4.5, Naive Bayes, and MLP and RBF neural networks. It is found that KNN can achieve more accurate classification for local regions where positives are underrepresented and that the local imbalance ratio and the local region size are important for the KNN performance. However the problem of how to improve KNN for imbalance learning is not addressed.

Orthogonal to imbalance classification, following the seminal work on the Nearest Neighbour classification [36], there have been extensive studies in the literature on improving the KNN algorithm: One class of studies, including for example [22, 17, 37, 38, 39, 40, 41], focus on selecting prototypes (data points) or learning new prototypes with the purpose of reducing noises in the training instance space. Another class of studies, for example [42, 43], focus on improving the posterior probability estimation from nearest neighbours through adjusted neighbourhoods or weighted distance computation. The purpose of all these existing studies is to improve the overall classification performance for KNN rather than class-specific precision in the presence of class imbalance.

3. Classification Decisions for the Rare

The KNN algorithm can be viewed as an empirical Bayes decision rule where $P(C_i|t)$, the posterior probability of query instance t for class C_i, is estimated from the k nearest neighbours of t. Instances are classified to the class with the highest posterior probability. For example for a two-class problem with the positive class C_+ and negative class C_-, an instance t is classified to the positive class if $P(C_+|t) > 0.5$, otherwise t is classified to the negative class. In many applications, it is also important to rank all instances for a class from the most probable to the least probable. A ranking function that produces ranking scores for instances is needed to ideally generate ranking of examples in accordance with their true class probability.

In the presence of skewed class distribution, negatives are frequent in many local regions, and the standard KNN classification decision based on posterior

class probability estimated from query neighbourhoods has an inherent bias towards the frequent class. To achieve better classification decisions, better strategies to calibrate the posterior positive probability are needed. Specifically two research questions need to be addressed:

- How should the local neighbourhood for query instances be decided so that it has adequate positive presence and is not overly enlarged?
- How to estimate the positive probability for query neighbourhoods to better calibrate and rank query instances in terms of their positive propensity? It is desired that the positive probability estimates can reliably rank query instances in accordance with their likelihood for being positive.

We propose to dynamically decide query neighbourhoods to ensure adequate positive presence locally. With standard KNN classification, very few or even zero positives are present in fixed-size regions of k instances. Given a query instance t, the value of k should be set locally and dynamically so that the local region has adequate positive frequency and on the other hand includes only closest neighbours of t that have posterior class probability approximately equal to that of t. We propose a simple heuristic strategy to set k dynamically based on local class distribution.

In this paper, different from existing studies [14, 13, 15], we propose to directly adjust the posterior probability estimation for query instances. With standard KNN, positive probability for query instances are estimated from their local neighbourhoods. The frequency-based approach can produce extreme probability estimates near zero when positives are very rare, and is not a reliable method for class probability estimation. Smoothing is a common approach to probability estimation. A de facto standard smoothing method is the Laplace estimate [44]. For a two-class problem, the positive class probability is estimated as $\frac{p+1}{p+n+C}$, where p and n are numbers positive and negative instances, and $C = 2$. The m-estimate [45] is another smoothing approach [46], where the positive class probability is estimated to be $\frac{p+b*m}{p+n+m}$, and m is a parameter that controls how much scores are shifted towards the positive prior in the training population.

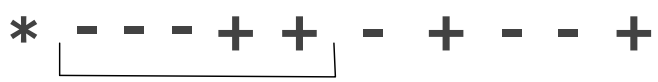
However setting m needs considerable computation. The smoothing approaches, either the Laplace estimate or the m-estimate, do not directly address the class imbalance issue. When positives are scarce, smoothing approaches generate close probability estimates near the set priors from the terms in the smoothing functions, and as a result can not distinguish and rank query instances in terms of their positive propensity.

Different from the smoothing approaches, we focus on obtaining calibrated positive probability estimates from the rare positives in local regions. We propose a simple, intuitive strategy based on the positive distribution in local neighbourhoods to adjust the class frequency terms p and n in the smoothing functions. The strategy results in calibrated positive probability estimation for query instances and is aimed to distinguish and rank them in terms of their true positive propensity. Our discussions next will be based on the Laplace estimate function but the same principle applies to the m-estimate.

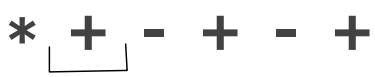
4. Forming Dynamic Query Neighbourhoods

In this section we describe our strategy of forming dynamic query neighbourhoods to increase the sensitivity of KNN for the rare class. For ease of discussions we will be based on the two-class problem, and the rare class is the *positive* class. As discussed later, our proposed approach can be easily generalised to multi-class problems.

On imbalanced datasets of within-class imbalance [47], for a standard KNN classifier, the $k(\geq 1)$ nearest neighbours for a query instance may contain few or even zero positives in its vicinity due to positive sample sparsity, and as a result the KNN classifier can not accurately characterise the positive propensity for the query instance. To rectify this situation, we enlarge the nearest neighbourhood for the query instance such that two conditions are satisfied: (1) the query neighbourhood contains at least k *positive* instances. and (2) the query neighbourhood limit reaches a positive-negative border, As a result, depending on the positive distribution surrounding the query instance, the query neigh-



(a) $r = 5, k' = 2$



(b) $r = 1, k' = 1$

Figure 1: An example dynamic local neighbourhood for a query instance (*) for $k = 1$.

bourhood has a varying size, and contains a varying number k' of positives, and $k' \geq k$. The varying neighbourhood size r and number of positives k' more accurately characterise the positive propensity for the query instance:

- If the local region centred at the query instance is sparsely populated with positives, it is very likely that r is large and k' is close to k. The query instance has low positive propensity.
- If the region contains relatively sparsely distributed positives, r is likely to be of moderate size and k' is still close to k. The query instance has moderate positive propensity.
- Otherwise the region is densely populated with positives. It is likely that r is small and $k' > k$. The query instance has high positive propensity.

As discussed in Section 7, experiments show that this dynamic local query neighbourhood formulation strategy can form dynamic query neighbourhoods and more accurately characterise the positive probability for the query instance, which sets the basis for accurate rare-class classification.

Fig. 1 shows an artificial two-class imbalanced problem, where positive (minority) training instances are denoted as "+", negative (majority) instances are denoted as "-", and "*" denotes the query instances. The diagrams are in one dimension so that it is easier to show the distance between instances and the size of different regions. As can be seen, for $k = 1$ and for a given query instance t (denoted as *), the query neighbourhood for classification varies depending on

the distribution of neighbouring positives:

- Fig. 1(a) shows the situation when the nearest neighbour of the query instance is negative. The neighbourhood for t is expanded to include the nearest positives until reaches the positive-negative border. As a result the neighbourhood is indeed the five-instance neighbourhood of t, or $r = 5$. Note also that due to the two positive instances are neighbours $k' = 2$.

- Fig. 1(b) shows the case when the nearest neighbour of the query instance is positive. As a result the neighbourhood for t reaches the positive-negative border, and $r = 1$ and $k' = 1$.

The dynamic query neighbourhoods of varying sizes provide us the benefit to determine whether to apply adjustment for estimating the positive posterior probability, as will be discussed in the next section.

5. Adjusting the Positive Posterior Probability Estimation

In this section we discuss our strategies for adjusting the positive posterior probability estimation for query neighbourhoods.

5.1. Contrasting local to global positive distributions

Let r be the number of instances in the local neighbourhood for a query instance t. Let q be a random variable of the class label for t, then the probability of q taking the positive label, or being a "success" follows the binomial distribution $B(r, q)$. The confidence interval for q based on the observed positive proportion $\hat{q} = k'/r$ is

$$\hat{q} \pm z_{\alpha/2}\sqrt{\hat{q}(1-\hat{q})/r}. \tag{1}$$

where $z_{\alpha/2}$ is the *z-score* for normal distribution corresponding to the confidence level of $100(1-\alpha)\%$. When the sample size r is small, which occurs often for local neighbourhoods, Equation 1 often understates the width of the true interval. A

correction to the estimation of the confidence interval for q for small sample size is as follows [48], where $z_0 = z_{\alpha/2}$:

$$\frac{\hat{q} + z_0^2/2r \pm z_0\sqrt{\hat{q}(1-\hat{q})/r + z_0^2/4r^2}}{1 + z_0^2/r} \tag{2}$$

Equation 2 is used to estimate the *local* positive confidence interval for query regions and Equation 1 is used to estimate the *global* positive confidence interval for the training population. We adjust the posterior positive probability estimation for a query instance towards the positive class if the lower endpoint of its local positive confidence interval is higher than the higher endpoint of the global positive confidence interval. The confidence level of $100(1-\alpha)\%$ is a parameter tuning the performance of our proposed approach. Note that when the local positive interval is comparable to the global positive interval then no adjustment will be applied to the positive posterior probability, and our algorithm will be reduced to standard KNN.

Generally a large number of samples mean the confidence interval estimation not very sensitive to confidence level settings. Assuming a sufficiently large training population, the confidence level for global positive confidence interval estimation, denoted as c_g, is generally set to high (90%) to get loose confidence intervals, which makes it hard to apply adjustment for classification decision. Confidence level settings for the query neighbourhood, denoted as c_r, from low to high implies small to large positive confidence intervals. Our experiments (c.f. Section 7.3) show that on imbalanced datasets higher c_r generally leads to more accurate rare-class classification.

5.2. Positive-biased posterior probability estimation

Given the neighbourhood for a query instance t comprising k' positive and n negative training instances respectively, the Laplace smoothing estimate for the positive probability of t, $\frac{k'+1}{k'+n+2}$, tends to be close to the prior of 0.5 when the query neighbourhood size $(k'+n)$ is very small. To address this issue, a modified Laplace estimate for posterior class probability commonly used [49].

Specifically, the positive posterior probability for t is estimated as:

$$P(C_+|t) = \frac{k' + \frac{1}{|D|}}{k' + n + \frac{2}{|D|}} \tag{3}$$

Note that in Equation 3 the class factors $\frac{1}{|D|}$ and $\frac{2}{|D|}$ are normalised by the training dataset size $|D|$ and reduces the strong bias towards 0.5.

On very skewed datasets, many local regions rarely have positives and very likely their positive probabilities estimated by Equation 3 are all close to the prior by the constant terms. A strategy is desired to compute the true positive propensity for these regions so as to distinguish and effectively rank the corresponding query instances. To this end, if the local positive interval for a query instance t(Equation 2) is higher than the global positive interval(Equation 1), intuitively it indicates that t has higher posterior positive probability than the positive prior based on the observed positive frequency in the training population. It is then reasonable to decide that t has higher positive propensity than that observed in the local positive frequency. The positive posterior probability estimation for t in Equation 3 therefore should be adjusted. Let λ denote the positive odds (P:N) in the query neighbourhood over the positive odds in the global population. As an example, if P:N=1:5 in the query region and P:N=1:10 in the global training population, then $\lambda = 2$. The positive posterior probability for query instance t is adjusted according to λ, as follows:

$$P(C_+|t) = \frac{k' + \frac{1}{|D|}}{k' + \frac{1}{\lambda} * n + \frac{2}{|D|}}, \tag{4}$$

Comparing Equations 3 and 4, λ takes into account the local versus global class imbalance levels and the positive odds ratio in the local neighbourhood versus the global population indicates the positive propensity for the query instance. Equation 4 is able to elegantly handle both imbalanced and balanced scenarios. When λ is extremely high, which indicates higher the local region has high propensity for the positive class, $P(C_+|t)$ by Equation 4 is much higher than that by Equation 3. When $\lambda = 1$, Equation 4 falls back to Equation 3.

Adjustment to posterior class probability estimation has been proposed in previous studies [50, 19], but for class probability estimation from the modified

class distribution after re-sampling. Weiss and Provost [50] proposed to adjust posterior probability estimation for a decision tree learnt from re-sampled training data. Specifically for a leaf node with p positive and n negative instances, the positive probability is estimated as $\frac{p+1}{p+o*n+2}$, where o is the positive over-sampling ratio. For example, if P:N=1:10 in the natural class distribution, but P:N=1:1 after re-sampling, then o is $1.0/0.1 = 10$, and the positive probability is estimated as $\frac{p+1}{p+10*n+2}$. In [19], an equivalent probability adjustment strategy is proposed, based on class fraction rather than class ratio after re-sampling. Different from these two studies, we aim to bias the posterior probability estimation towards the frequent class based on the natural class distribution in local neighbourhoods. To the best of our knowledge, Equation 4 is the first attempt to make use of the natural class skewness level of training data to bias classification decision.

Generally for a multi-class problem where there are one or several rare classes, the one-versus-others scheme can be applied to decompose the problem into multiple binary classification problems. For each binary classification problem, our proposed posterior adjustment approach can be applied to estimate the posterior probability for each rare class. Given the rare occurrences of rare class instances in any query neighbourhood, the posterior probability estimation for classes shall not be distorted.

6. The KRNN algorithm

We propose the *k Rare-class Nearest Neighbour* (KRNN) algorithm (Algorithm 1) that applies all strategies discussed in Section 5. Input for the algorithm includes a given query instance t, the training population T, a given minimal number of positive nearest neighbours k, the global confidence level c_g and the local confidence level c_r. KRNN outputs $P(C_+|t)$, the estimated positive posterior probability for the query instance t.

As discussed in Section 4, for a query instance t, the training instance space is searched for the local region R centred at t that contains at least k positives. As

Algorithm 1 The KRNN classification algorithm

Input:

a) Training set D;

b) Query instance t;

c) The minimal number of positive nearest neighbours k;

d) The global confidence level c_g and local confidence level c_r

Output: Positive posterior probability for t

1: $R \leftarrow$ the neighbourhood for t with $k' \geq k$ positive nearest neighbours

2: $r \leftarrow |R|$

3: $L_g \leftarrow$ interval by Equation 1 from D, positive frequency in D, and c_g

4: $L_r \leftarrow$ interval by Equation 2 from R, positive frequency in R ($\frac{k'}{r}$), and c_r {*positive posterior probability estimation adjustment*}

5: **if** (L_r.low-end $> L_g$.high-end) **then**

6: $\lambda \leftarrow \frac{\text{P:N in R}}{\text{P:N in D}}$

7: $P(C_+|t)$ is computed using Equation 4 from λ, r, k', and $|D|$

8: **else**

9: $P(C_+|t)$ is computed using Equation 3 from r, k' and $|D|$

10: **end if**

there may exist clusters of positive instances of more than one positive instance, the search may result in that R contains k' positives, and $k' > k$. Given the size r for the query neighbourhood R, the positive posterior probability estimation is adjusted according to the positive bias factor λ (Line 7) and Equation 4. For a given k, the size of dynamic neighbourhood R varies significantly depending on the positive distribution in R. Our experiments show that, with $k = 1$, the query neighbourhood for classification can contain from one instance to hundreds of instances. The dynamic neighbourhoods accurately characterise the positive distribution profile in local neighbourhoods for query instances, and is important for KRNN to achieve good performance in the Receiver Operating Characteristic (ROC) space (See experiment results in Section 7.3).

Similar to the standard KNN model, the choice of k for KRNN can be

determined empirically by cross validation. A large k value may lead to a large nearest neighbourhood containing training instances far from the query instance that have posterior probability different from the query instance. Due to the data sparsity associated with imbalanced datasets [18], a large k value for KRNN often results in extremely large neighbourhoods for classification. Our experiments show that $k = 1..3$ typically gives good classification performance. The default setting is $k = 1$. Experiments on settings of k are presented in Section 7.3.

Parameters c_g and c_r in KRNN are confidence levels for estimating the global and local positive confidence intervals High confidence levels c_g and c_r result in wider confidence intervals and Equation 4 is more likely applied, which generally leads to high recall for the positive class. On the contrary, low confidence levels result in tight intervals and Equation 4 are less likely applied, which can lead to high precision for the positive class. By default c_g and c_r are set to 0.9.

7. Experiments

We conducted extensive 10-fold cross validation experiments to evaluate the performance of KRNN. Thirty imbalanced datasets, including 14 real-world and 16 artificial datasets are used for evaluation of classifiers. All classifiers were developed using the WEKA (Version 3.6.5) data mining toolkit [49].

7.1. Algorithm settings, evaluation metrics and datasets

KRNN is benchmarked against other imbalanced classification algorithms and strategies, including: a) the specificity-oriented learning algorithm KNN, or the IBk algorithm in WEKA, and three recent algorithms in the KNN family for imbalanced classification, namely ENN [13], PNN [15] and CCW-KNN [14]; b) generality-oriented learning algorithms J48, the C4.5 decision tree in WEKA, and LibSVM [51]; c) oversampling strategy SMOTE [7] and cost-sensitive learning strategy MetaCost [8], using IBk, J48 and LibSVM (Version 3.18) as base learners. Settings of the algorithms are as follows:

- For KRNN, IBk, ENN and CCW-KNN in the KNN family, $k = 1$. For KRNN $c_g = 0.9$ and $c_r = 0.9$. For CCW-KNN, the instance weighting is set as "Weight by class confidence and 1/distance".

- For the generality-oriented learning algorithms, J48 is set with the "-M1" option (minimum one instance is allowed for a leaf node without pruning). LibSVM settings are: the RBF kernel, cost = 10, gamma = 0.001, and "estimate class probabilities". These settings ensure LibSVM has good performance on most datasets.

- Settings for SMOTE and MetaCost imbalanced learning strategies: SMOTE minority over-sampling of 3 times more instances for the minority class. With MetaCost [8] mis-classification cost for the positive class is set to the negative-to-positive ratio in the training population.

The ROC curve [52] has been widely used for evaluating imbalanced classification performance, as the curve is irrespective to class distribution. The Area Under ROC (AUC) as an aggregate metric measuring the overall classification performance of classifiers for the rare class across different false positive rates. Precision and recall at a specific classification threshold (typically 50%) are also widely to evaluate class-specific classification accuracy, which are defined as:

$$Precision = \frac{TP}{TP + FP},\ Recall = \frac{TP}{TP + FN},\ F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

where TP, FP, FN represent numbers for true positives, false positives, and false negatives respectively. The ROC convex hull [2] highlights classification accuracy at different levels of sensitivity [24, 52] and has been used for evaluating imbalanced classification [7]. In the ROC space, if a point falls on the convex hull for a set of points the corresponding classifier is potentially an optimal classifier for the FP rate.

Fourteen real-world imbalanced datasets from various domains are used in our experiments, as summarised in Table 1. Datasets range from highly imbal-

[2] Available at `http://home.comcast.net/~tom.fawcett/public_html/ROCCH/`.

Table 1: Fourteen real-world imbalanced datasets, ordered in increasing positive frequency.

Dataset	Size	#Attr	Class (Pos, Neg)	P:N (Pos frequency)
Hiva*	3844	1617	(1, -1)	135:3709 (3.51%)
Oil	937	47	(true, false)	41:896 (4.38%)
Hypo-thyroid	3163	25	(true, false)	151:3012 (4.77%)
Sylva*	13085	217	(1, -1)	805: 12280 (6.15%)
PC1ˆ	1109	21	(true, false)	77:1032 (6.94%)
Glass	214	9	(3, other)	17: 197 (7.94%)
Satimage	6435	36	(4, other)	626: 5809 (9.73%)
CM1ˆ	498	21	(true, false)	49: 449 (9.84%)
KC1ˆ	2109	21	(true, false)	326: 1783 (15.46%)
SPECT_F	267	44	(0, 1)	55: 212 (20.60%)
Hepatitis	155	19	(1, 2)	32: 123 (20.65%)
Vehicle	846	18	(van, other)	199: 647 (23.52%)
Ada*	4146	49	(1, -1)	1029: 3117 (24.82%)
German	1000	20	(2, 1)	300: 700 (30.00%)

*: Agnostic learning datasets; ˆ : Software engineering datasets.

anced (with a positive frequency of 3.51%) to lowly imbalanced (with a positive frequency of 30.00%). Seven datasets from the UCI Machine Learning repository [53] are selected. With multi-class datasets, one class is chosen as the positive and the others are the negatives, as shown in the fourth column of Table 1. For the SPECT_F dataset the version with numeric attributes is used. Datasets PC1, CM1 and KC1 are software engineering datasets for predicting software defects, obtained from the NASA IV&V Facility Metrics Data Program (MDP) repository [3]. These datasets have been widely used in software engineering research [3] for software defect prediction. Generally only a minority of modules (around 10% on average) contain defects. The Oil dataset was provided by Robert Holte [1], and the task is to predict the oil spill from satellite images, which generally only occurs in 41 out of 937 samples (4.3%). The dataset has been widely used in imbalanced learning research. Hiva, Sylva and Ada are downloaded from the Workshop on Agnostic Learning versus Prior

[3] http://mdp.ivv.nasa.gov/index.html

Table 2: Overall AUC results: KRNN versus KNN family of algorithms. Highest AUC in bold ("<0.001" deemed equal). p-value < 0.05 in bold for statistical significance (versus KRNN).

Dataset	KRNN	PNN	ENN	CCW-KNN	IBk	SMOTE-IBk	MetaCost-IBk
Hiva	**0.825**	0.810	0.670	0.715	0.661	0.674	0.708
Oil	**0.875**	0.865	0.742	0.849	0.721	0.752	0.750
Hypo-thyroid	**0.942**	0.932	0.771	0.871	0.789	0.796	0.821
sylva	**0.994**	0.986	0.949	0.974	0.914	0.957	0.933
PC1	**0.840**	0.833	0.820	0.821	0.740	0.730	0.791
Glass	**0.752**	0.750	0.630	0.581	0.603	0.719	0.570
Satimage	**0.963**	0.954	0.866	0.918	0.829	0.879	0.851
CM1	**0.715**	**0.716**	0.600	0.703	0.589	0.602	0.625
KC1	0.771	0.761	0.754	**0.824**	0.735	0.760	0.775
SPECT_F	0.743	**0.745**	0.669	0.706	0.619	0.739	0.664
Hepatitis	0.801	0.698	0.698	**0.850**	0.678	0.645	0.698
Vehicle	**0.983**	0.969	0.897	0.967	0.916	0.934	0.922
Ada	**0.826**	0.807	0.726	0.784	0.692	0.691	0.712
German	**0.736**	0.712	0.657	0.734	0.660	0.665	0.660
Average	**0.840**	0.824	0.746	0.807	0.725	0.753	0.749
p-value	–	**0.037**	**<0.001**	**0.048**	**<0.001**	**<0.001**	**<0.001**

Knowledge Challenge & Data Representation [4].

A collection of 16 artificial 2-dimensional datasets of two classes are constructed using the experimental settings in [54]. For each dataset, instances are randomly generated according to the normal distribution. For the positive class mean and standard deviation are set to 0 and 1 for each dimension, that is, $\mu_+ = [0, 0]$ and $\sigma_+ = [1, 1]$. For the negative class, mean and standard deviation are set to 2 for one dimension, and to 0 and 2 respectively for the other dimension, that is, $\mu_- = [2, 0]$ and $\sigma_- = [2, 2]$. Such settings form overlapping classes along both dimensions. The imbalance level of the 16 datasets varies – the positive frequency ranges from 50% to 5.88%. Specifically, the number of positive instances in all datasets is set to a fixed number of 50, whereas the number of negatives increases from 50 to 800 in increments of 50.

[4] `http://www.agnostic.inf.ethz.ch/datasets.php`

Table 3: Overall AUC results: KRNN versus generality-oriented models. Highest AUC in bold ("<0.001" deemed equal). p-value < 0.05 in bold for statistical significance (versus KRNN)

.

				SMOTE		MetaCost	
Dataset	KRNN	J48	SVM	J48	SVM	J48	SVM
Hiva	**0.825**	0.625	0.775	0.656	0.785	0.685	0.802
Oil	0.875	0.685	0.885	0.757	**0.917**	0.764	**0.917**
Hypo-thyroid	0.942	0.924	0.980	0.916	**0.981**	0.937	**0.982**
sylva	0.994	0.966	**0.999**	0.971	**0.999**	0.979	**0.999**
PC1	**0.840**	0.789	0.795	0.768	0.777	0.760	0.704
Glass	0.752	0.696	0.697	**0.880**	0.851	0.864	0.622
Satimage	**0.963**	0.767	0.919	0.766	0.880	0.816	0.900
CM1	0.715	0.607	**0.762**	0.638	0.725	0.668	0.710
KC1	0.771	0.640	0.687	0.642	**0.793**	0.695	0.787
SPECT_F	0.743	0.626	0.822	0.622	**0.847**	0.643	0.806
Hepatitis	0.801	0.753	0.839	0.645	0.862	0.746	**0.873**
Vehicle	**0.983**	0.921	**0.983**	0.924	0.895	0.929	0.873
Ada	0.826	0.756	0.867	0.770	**0.870**	0.758	0.838
German	0.736	0.608	**0.743**	0.616	0.708	0.620	0.741
Average	0.840	0.740	0.840	0.755	**0.849**	0.776	0.825
p-value	–	**<0.001**	0.943	**0.002**	0.602	**0.003**	0.426

7.2. The overall performance of KRNN

We evaluated the performance of KRNN by ten-fold cross validation experiments in WEKA. Two-tailed paired t-tests were used to test the statistical difference between results. AUC was used to evaluate the overall classification performance of KRNN. Table 2 shows the AUC results for KRNN in comparison with other models in the KNN family, namely PNN, ENN, CCW-KNN, SMOTE-IBk and MetaCost-IBk. For fair comparison $k = 1$ for all models. Overall KRNN has the highest average AUC value of 0.840 and this result is significantly better than all other models consistently ($p < 0.05$). Especially KRNN has significantly better average AUC than that for SMOTE-IBk and MetaCost-IBk — 0.840 versus 0.753 and 0.749. This result highlights that the local strategies in KRNN are more effective than the global strategies of resampling and cost-sensitive learning for rare-class classification. Table 3 shows the AUC result of KRNN against popular generality-oriented models the deci-

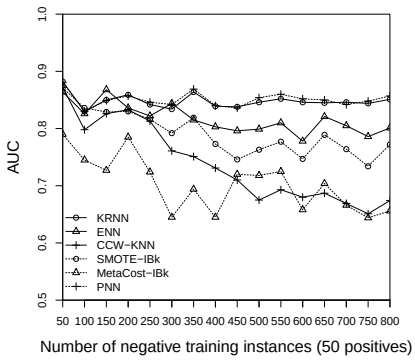
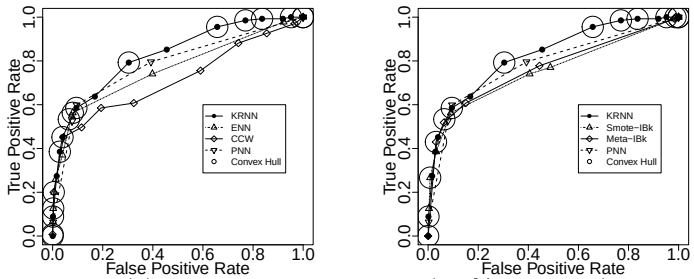


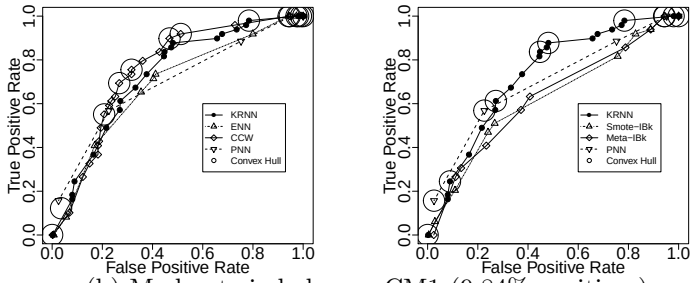
Figure 2: AUC results for KRNN vs. other KNN family algorithms on artificial datasets.

sion and support vector machine, and meta-learning strategies with these models, namely the J48, LibSVM, SMOTE-J48, SMOTE-SVM, MetaCost-J48 and MetaCost-SVM. KRNN achieves an average AUC (0.840) significantly higher than that of J48 (AUC = 0.740) , as well as SMOTE-J48 (0.755) and MetaCost-J48 (0.776) with $p < 0.05$. KRNN has an average AUC (0.840) comparable with, or not statistically different from, that for SVM (0.840) , SMOTE-SVM (0.849) and MetaCost-SVM (0.825).

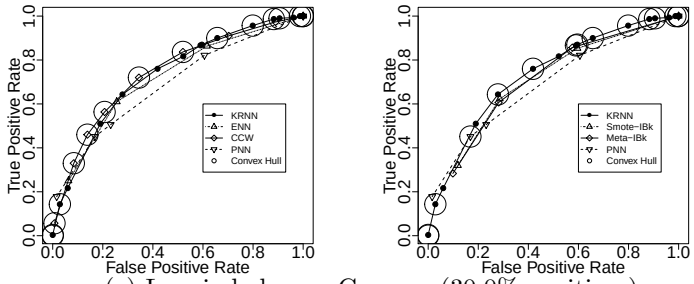
We further evaluated KRNN against other models in the KNN family on the 16 artificial datasets, and results are plotted in Fig. 2. In the figure, the leftmost data point has perfect balance where P:N=50:50, whereas the rightmost point is highly imbalanced with P:N=50:800. It can be seen that in general at all imbalance levels KRNN and PNN show the best and stable performance among the six models. KRNN has an average AUC of 0.848 with a range of 0.882 (P:N=50:50) to 0.851 (P:N=50:800) whereas PNN has an average AUC of 0.851 with a range of 0.881 (P:N=50:50) to 0.857 (P:N=50:800). ENN and SMOTE-IBk also show lower but relatively stable AUC performance across different imbalance levels. In contrast CCW-KNN and MetaCost-IBk show degrading performance with increasing imbalance levels: MetaCost-IBk especially shows oscillating performance with respect to imbalance levels. This partly shows that the cost matrix setting for MetaCost according to the inverse of class ratio has



(a) High imbalance: Hiva (3.5% positives)



(b) Moderate imbalance: CM1 (9.84% positives)



(c) Low imbalance: German (30.0% positives)

Figure 3: ROC convex hull: KRNN versus other imbalanced classification models.

not been effective when the negative content is extremely high.

We analysed the convex hull for models in the KNN family, which characterise in details the performance of models. Fig. 3 shows the ROC curves for KRNN against five other models where the points on the convex hull are highlighted in circles. For readability curves are plotted in two graphs for three representative datasets at different imbalance levels: Hiva of high imbalance (3.5% positives), CM1 of moderate imbalance (9.84% positives) and German of low imbalance (30% positives).

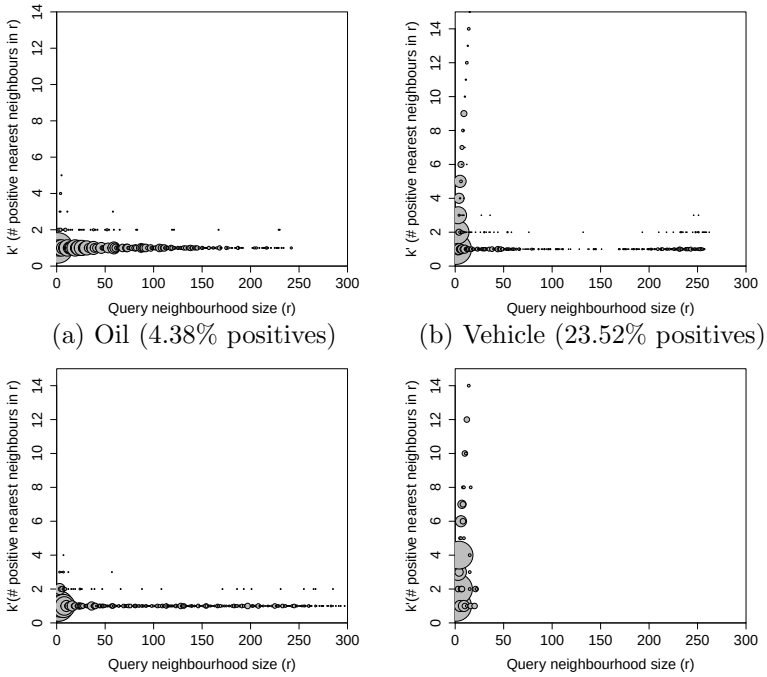
Graphs on the left plot KRNN against PNN, and ENN and CCW-KNN. On the highly imbalanced dataset Hiva (3.51% positives), KRNN dominates the ROC space at all FP rates, which is consistent with KRNN having the highest AUC of 0.825 (Table 2). On the moderately imbalanced dataset CM1, CCW-KNN does not have the highest AUC but has the largest number of points on the convex hull, mostly at high FP rates. Still at the very low FP rate of 0.02 and 0.08, KRNN and PNN lie on the convex hull. KRNN and PNN are obviously good choices for those applications where accurate prediction at low FP rate is highly desirable. In the left graph of Fig. 3(c), on the German dataset of low imbalance (30% positives), KRNN and CCW-KNN have comparable AUC results (0.736 versus 0.734), and almost equal appearances on the Convex Hull. Still at low FP rates, more points from KRNN lie on the ROC convex hull, which again shows that KRNN is a strong model when low FP rate is desired.

Graphs on the right of Fig. 3 plot KRNN and PNN against SMOTE-IBk and MetaCost-IBk. Clearly points of the KRNN and PNN dominate the convex hull at all misclassification costs, and on all datasets from high to low imbalance levels. For example on the very imbalanced Hiva dataset, KRNN has seven out of 14 data points on the convex hull. In contrast SMOTE-IBk and MetaCost-IBk each has only two of eight and two of seven points on the convex hull. Overall the graphs on the right of Fig. 3 (a)–(c) convincingly demonstrate that the local query neighbourhood and probability adjustment strategy of KRNN and PNN significantly outperforms the re-sampling and cost-sensitive learning strategies for the KNN model.

7.3. The properties and parameters of KRNN

We analysed the properties and parameters of KRNN, including the dynamic query neighbourhood size r (Section 3), the positive-biased posterior probability estimation strategy (Section 5.2), the settings of confidence levels c_g and c_r and the setting of k.

The dynamic query neighbourhood for classification: We examined the dynamic query neighbourhood size r and the number of positive samples k' for



(a) Oil (4.38% positives) (b) Vehicle (23.52% positives)

(c) P50:N800 (d) P50:N50

Figure 4: The distribution of dynamic query neighbourhoods

$k = 1$ on real-world and artificial datasets of high and low imbalance. Fig. 4 plots distribution of query neighbourhoods of different sizes in the (r, k') space. Each point in the (r, k') space represents a neighbourhood of r instances where k' are positives, and the size of the circle at the point represents the number of local neighbourhoods falling onto the point. Obviously the distribution of neighbourhoods in Fig. 4 (a) and (c) is very different from that in Fig. 4 (b) and (d). In Fig. 4(a) and Fig. 4(c), for highly imbalanced datasets Oil and P50:N800, the circles of different sizes spread along the x-axis ($r = 1..297$). In contrast in Fig. 4(b) and Fig. 4(d), for moderately imbalanced dataset Vehicle and balanced dataset P50:N50, large circles are mainly located on the low end of the x-axis Indeed 22% of query neighbourhoods for Vehicle and 49% of query neighbourhoods for P50:N50 have $r \leq 4$. We can see that the dynamic query neighbourhood size r varies significantly on severely imbalanced datasets due to

within-class imbalance and data sparsity, as reported in [18] and [2].

The distribution of positives k' in query neighbourhoods also varies. It can be seen that in Fig. 4(a) and Fig. 4(c), on highly imbalanced datasets, circles mostly concentrate in the area where $k' = 1$. In contrast in Fig. 4(b) and Fig 4(d), on moderately balanced datasets, circles spread along the y axis and has a wide range of k' values. In Fig. 4(a) and Fig. 4(c), on the highly imbalanced datasets Oil and P50:N800, $k' = 1$ for most query neighbourhoods, even though r varies significantly. On the Oil dataset of 937 training instances, for 930 neighbourhoods $k' = 1$ but r varies in 1..230. In Fig. 4(b) and Fig. 4(d) on the moderately imbalanced dataset Vehicle and balanced dataset P50:N50, although a large number of query neighbourhoods typically have small r values (lying on the low end of the x-axis), for a given r, k' varies and spreads along the y-axis. For example on Vehicle, $r = 3$ for 49 query neighbourhoods, and k' varies from 1 to 3, for 17, 7 and 25 neighbourhoods respectively.

In summary the varying distribution of k' within the dynamic query neighbourhoods of KRNN for datasets of different imbalance level forms an accurate characterisation of the positive distribution profile for query instances. The strategy of dynamic query neighbourhood of KRNN forms the basis for accurate rare classification performance at different positive sensitivity settings.

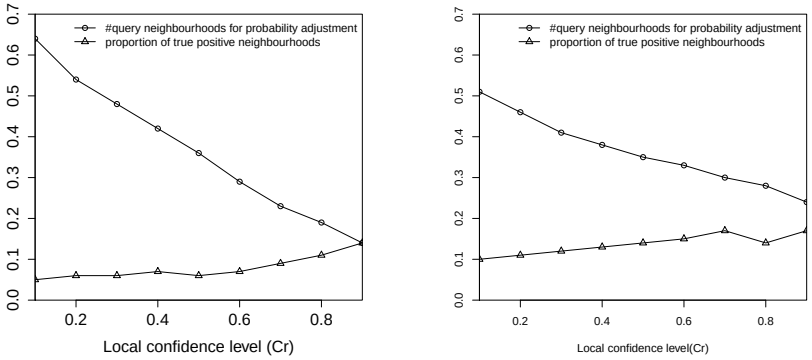
The adjusted posterior probability estimation strategy: KRNN applies the adjusted positive probability estimation Equation 4 when the local neighbourhood has higher positive content or otherwise the standard Laplacian estimation Equation 3. This adaptive positive probability estimation strategy can achieve significantly higher recall than always Equation 3 strategy and higher precision than the always Equation 4 strategy. Table 4 shows the results for Precision, Recall and F_1 for these three strategies. KRNN significantly improves the recall for Equation 3 of standard Laplacian estimation (p-value = 0.003), with an average recall of 0.729 versus 0.657, while maintaining comparable average precision (0.385 versus 0.436). On the other hand KRNN also significantly improves the precision for Equation 4 (p-value = 0.024), with an average precision of 0.385 versus 0.381, while maintaining comparable recall (0.729 versus 0.745). We fur-

Table 4: The Recall, Precision and F_1 for posterior probability estimation strategies

	KRNN			Equation 3			Equation 4		
Dataset	Prec	Recall	F_1	Prec	Recall	F_1	Prec	Recall	F_1
Hiva	0.188	0.516	0.276	0.282	0.519	0.365	0.158	0.615	0.251
Oil	0.265	0.659	0.378	0.418	0.561	0.479	0.265	0.659	0.378
Hypo-thyroid	0.381	0.755	0.506	0.619	0.689	0.652	0.381	0.755	0.506
Sylva	0.627	0.983	0.766	0.716	0.950	0.817	0.627	0.983	0.766
PC1	0.231	0.701	0.347	0.313	0.532	0.394	0.231	0.701	0.347
Glass	0.250	0.588	0.351	0.250	0.353	0.293	0.250	0.588	0.351
Satimage	0.452	0.917	0.606	0.540	0.843	0.658	0.452	0.917	0.606
CM1	0.215	0.490	0.299	0.189	0.347	0.245	0.195	0.480	0.277
KC1	0.315	0.632	0.420	0.361	0.574	0.443	0.315	0.632	0.420
SPECT_F	0.357	0.836	0.500	0.357	0.818	0.497	0.329	0.909	0.483
Hepatitis	0.468	0.688	0.557	0.361	0.574	0.443	0.449	0.688	0.543
Vehicle	0.772	0.970	0.860	0.778	0.970	0.863	0.761	0.975	0.855
Ada	0.486	0.748	0.589	0.488	0.746	0.590	0.482	0.765	0.591
German	0.436	0.727	0.545	0.436	0.727	0.545	0.438	0.760	0.556
Average	**0.385**	**0.729**	0.500	0.436	0.657	0.520	0.381	0.745	0.495
p-value				0.057	**0.003**	0.287	**0.024**	0.093	0.093

ther evaluated the AUC results for KRNN against the approaches of Equation 3 and Equation 4. We found that the three approaches achieve comparable AUC results of 0.840, 0.840 and 0.841 respectively. This result can be explained by that even though KRNN has more accurate estimation for the positive probability, as evidenced by the higher recall than Equation 3 and higher precision than Equation 4, it does not significantly change the relative ranking of positive and negative instances compared to Equation 3 or Equation 4.

The confidence levels c_g and c_r: In the KRNN algorithm, the confidence levels c_g and c_r control the confidence intervals for the estimated positive probability. We conducted experiments to evaluate how confidence level settings catch true positive neighbourhoods. Fig. 5 plots the number (normalised by training population size) of neighbourhoods with probability adjustment and the proportion of true positive neighbourhoods when $c_g = 0.9$ and c_r varies from 0.1 to 0.9. Generally it can be seen from the figure, on both the real-world and artificial datasets, that higher c_r values results in a smaller number of query

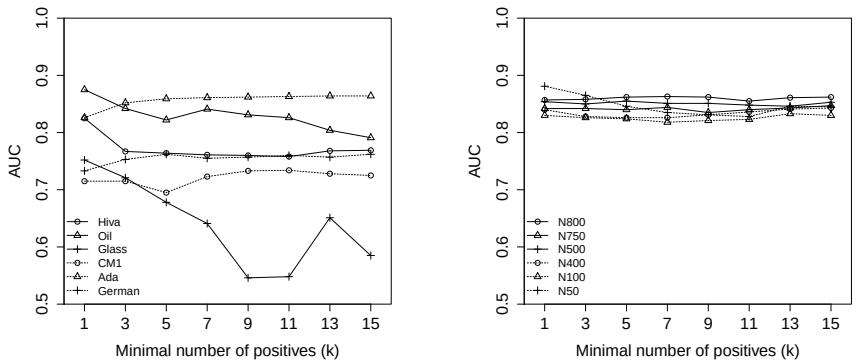


(a) On the real-world dataset Oil (b) On artificial dataset P50:N800

Figure 5: Confidence level settings and neighbourhoods with posterior probability adjustment

neighbourhoods having positive probability adjustment but a higher proportion of true positive neighbourhoods. In Fig. 5 (a), on the highly imbalanced Oil dataset, when c_r increases from 0.1 to 0.9, the number of query neighbourhoods where the probability adjustment is applied significantly decreases from 0.64 (598 of 937 instances) to 0.14 (131 of 937 instances), and the proportion of true positive query neighbourhoods increases from 0.05 to 0.14. In Fig. 5 (b), on the extremely imbalanced artificial dataset P50:N800, similar trends are observed. Our experiments confirmed that higher confidence levels lead to more true positive neighbourhoods have their positive probability adjusted towards the positive and therefore lead to more accurate rare-class classification.

The setting of k: The setting of k for KRNN specifies the minimal number of positives in a query neighbourhood for making classification decision. Recall that the due to dynamic neighbourhood formulation strategy, the final number of positives in the neighbourhood for classification can be higher, that is $k' > k$. We experimented on six real-world and six artificial datasets of high-, medium- and low-level of imbalance to evaluate the AUC performance of KRNN for different settings of k, $k = 1, 3, ..., 15$, and results are shown in Fig. 6. On the real-world datasets (Fig. 6 (a)) AUC is more sensitive to the varying k values. Moreover, AUC shows different trends on datasets of different imbalance levels. For example on the highly imbalanced dataset Hiva (3.5% positive), AUC ranges



(a) On real-world datasets (b) On artificial datasets (50 positives)

Figure 6: AUC results with settings of k

from 0.825 ($k = 1$) to AUC = 0.769 ($k = 15$), while on the low-imbalance dataset German (30% positive), AUC increases from 0.733 ($k = 1$) to 0.762 ($k = 15$). Generally for standard KNN classification, the setting of $k > 1$ for classification is mainly to reduce variance in class probability estimation so that classification decision is close to Bayes decision [36]. For KRNN, on imbalanced datasets, the setting of $k = 1$ can imply a large query neighbourhood due to severe within-class data sparsity [47, 2] and $k' > k$, and consequently the positive posterior probability is accurately estimated. So by default k is set to 1 for KRNN.

8. Conclusion and Future Work

Although resampling and cost-sensitive learning can improve the imbalanced classification performance for generality-oriented learning models like the decision tree, such global strategies do not have profound effect on specificity-oriented learning models like KNN. Based on this observation, we proposed k Rare-class Nearest Neighbour, or KRNN, that applies several strategies adjusting the induction bias of KNN in local neighbourhoods: 1) The local neighbourhood for classification decision is dynamically formed. 2) The posterior probability estimation for query instances is carefully adjusted and biased towards the rare class. These strategies more accurately characterise the rare-

class distribution for accurate classification. Extensive experiments on real-world and artificial imbalanced datasets showed that KRNN often significantly outperformed re-sampling and cost-sensitive learning strategies for imbalanced classification. KRNN also significantly outperformed recent specificity-oriented imbalanced classification algorithms in the KNN family.

For future work we will investigate extending KRNN to problems where there are multiple rare classes. We will also investigate extending our approach of estimating posterior class probability to applications where it is not enough to predict only the most likely class but also to rank instances based on class probability or to correctly estimate the true class probability.

Acknowledgements

The authors thank Dr. Wei Liu for the CCW-KNN code. This research is supported in part by an Australian Research Council Linkage Project (LP120200128). The authors thank the anonymous reviewers for their helpful comments.

References

[1] M. Kubat, R. Holte, S. Matwin, Machine learning for the detection of oil spills in satellite radar images, Machine Learning 30 (2-3) (1998) 195–215.

[2] G. M. Weiss, Mining with rarity: a unifying framework, SIGKDD Explorations 6 (1) (2004) 7–19.

[3] T. Menzies, J. Greenwald, A. Frank, Data mining static code attributes to learn defect predictors, IEEE Trans. on Software Engineering 33 (2007) 2–13.

[4] T. Fawcett, F. J. Provost, Adaptive fraud detection, Data Mining and Knowledge Discovery 1 (3) (1997) 291–316.

[5] Q. Yang, X. Wu, 10 challenging problems in data mining research, International Journal of Information Technology & Decision Making 5 (04) (2006) 597–604.

[6] N. V. Chawla, Data mining for imbalanced datasets: An overview, in: Data Mining and Knowledge Discovery Handbook, Springer, 2010, pp. 875–886.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.

[8] P. Domingos, MetaCost: A general method for making classifiers cost-sensitive, in: Proceedings of KDD 1999, 1999, pp. 155–164.

[9] N. Thai-Nghe, Z. Gantner, L. Schmidt-Thieme, Cost-sensitive learning methods for imbalanced data, in: Proceeding of 2010 International Joint Conference on Neural Networks (IJCNN10), 2010.

[10] P. Cao, D. Zhao, O. Zaiane, An optimized cost-sensitive SVM for imbalanced data learning, in: Proceedings. of PAKDD 2013, 2013, pp. 280–292.

[11] R. C. Holte, L. Acker, B. W. Porter, Concept learning and the problem of small disjuncts, in: Proceedings of IJCAI, 1989, pp. 813–818.

[12] K. Ting, The problem of small disjuncts: its remedy in decision trees, in: Proceedings of the 10th Canadian Conference on AI, 1994, pp. 91–97.

[13] Y. Li, X. Zhang, Improving k nearest neighbor with exemplar generalization for imbalanced classification, in: Proceedings of PAKDD, 2011, pp. 1–12.

[14] W. Liu, S. Chawla, Class confidence weighted kNN algorithms for imbalanced data sets, in: Proceedings of PAKDD, 2011, pp. 345–356.

[15] X. Zhang, Y. Li, A positive-biased nearest neighbour algorithm for imbalanced classification, in: Proc. of PAKDD 2013, Springer, 2013, pp. 293–304.

[16] V. Vapnik, L. Bottou, Local algorithms for pattern recognition and dependencies estimation, Neural Computation 5 (6) (1993) 893–909.

[17] D. W. Aha, D. Kibler, M. K. Albert, Instance-based learning algorithms, Machine learning 6 (1) (1991) 37–66.

[18] N. Japkowicz, Concept-learning in the presence of between-class and within-class imbalances, in: Advances in Artificial Intelligence, Springer, 2001, pp. 67–77.

[19] C. Elkan, The foundations of cost-sensitive learning, in: Proceedings of IJCAI, 2001, pp. 973–978.

[20] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[21] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.

[22] D. L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Transactions on Systems, Man and Cybernetics SMC-2 (3) (1972) 408–421.

[23] P. Flach, J. Hernandez-Orallo, C. Ferri, A coherent interpretation of AUC as a measure of aggregated classification performance, in: Proceedings of ICML, 2011, pp. 657–664.

[24] F. J. Provost, T. Fawcett, Robust classification for imprecise environments, Machine Learning 42 (3) (2001) 203–231.

[25] M. J. Pazzani, C. J. Merz, P. M. Murphy, K. Ali, T. Hume, C. Brunk, Reducing misclassification costs, in: Proc. of ICML, 1994, pp. 217–225.

[26] Y. Sun, M. S. Kamel, A. K. C. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognition 40 (12) (2007) 3358–3378.

[27] H. Dubey, V. Pudi, Class based weighted k-nearest neighbor over imbalance dataset, in: Advances in Knowledge Discovery and Data Mining, Springer, 2013, pp. 305–316.

[28] R. Akbani, S. Kwek, N. Japkowicz, Applying support vector machines to imbalanced datasets, in: Proceedings of ECML, Springer, 2004, pp. 39–50.

[29] H. Alhammady, K. Ramamohanarao, Using emerging patterns and decision trees in rare-class classification, in: Fourth IEEE International Conference on Data Mining (ICDM'04), IEEE, 2004, pp. 315–318.

[30] H. He, Y. Bai, E. A. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, in: Proceedings of IJCNN, 2008, pp. 1322–1328.

[31] G. M. Weiss, K. McCarthy, B. Zabar, Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs, in: Proceedings of ICDM, 2007, pp. 35–41.

[32] D. R. Carvalho, A. A. Freitas, A genetic-algorithm for discovering small-disjunct rules in data mining, Appl. Soft Comput. 2 (2) (2002) 75–88.

[33] D. Carvalho, A. Freitas, A hybrid decision tree/genetic algorithm method for data mining, Information Sciences 163 (1-3) (2004) 13–35.

[34] X. Hong, S. Chen, C. J. Harris, A kernel-based two-class classifier for imbalanced data sets, IEEE Transactions on Neural Networks 18 (1) (2007) 28–41.

[35] V. García, R. A. Mollineda, J. S. Sánchez, On the k-NN performance in a challenging scenario of imbalance and overlapping, Pattern Analysis and Applications 11 (3-4) (2008) 269–280.

[36] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1) (1967) 21–27.

[37] D. W. Aha, Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms, International Journal of Man-Machine Studies 36 (2) (1992) 267–287.

[38] D. R. Wilson, T. R. Martinez, Reduction techniques for instance-based learning algorithms, Machine learning 38 (3) (2000) 257–286.

[39] E. Pekalska, R. P. W. Duin, P. Paclík, Prototype selection for dissimilarity-based classifiers, Pattern Recognition 39 (2) (2006) 189–208.

[40] Y. Huang, C. Chiang, J. Shieh, E. Grimson, Prototype optimization for nearest-neighbor classification, Pattern Recognition 35 (6) (2002) 1237 – 1245.

[41] Y. Wu, K. Ianakiev, V. Govindaraju, Improved k-nearest neighbor classification, Pattern Recognition 35 (10) (2002) 2311 – 2318.

[42] J. Wang, P. Neskovic, L. Cooper, Neighborhood size selection in the k-nearest-neighbour rule using statistical confidence, Pattern Recognition 39 (2006) 417–423.

[43] C. Y. Zhou, Y. Q. Chen, Improving nearest neighbor classification with cam weighted distance, Pattern Recognition 39 (4) (2006) 635 – 645.

[44] F. Provost, P. Domingos, Well-trained pets: Improving probability estimation trees, Tech. Rep. CDER 00-04-IS, Stern School of Business, New York University (2000).

[45] B. Cestnik, Estimating probabilities: a crucial task in machine learning., in: ECAI, Vol. 90, 1990, pp. 147–149.

[46] B. Zadrozny, C. Elkan, Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers, in: ICML, Vol. 1, Citeseer, 2001, pp. 609–616.

[47] N. Japkowicz, Concept-learning in the presence of between-class and within-class imbalances, in: Proc. of Canadian Conference on AI, 2001, pp. 67–77.

[48] R. V. Hogg, E. Tanis, Probabilty and Statistical Inference, 8th Edition, Pearson Education, 2010.

[49] I. H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition, Morgan Kaufmann, 2011.

[50] G. M. Weiss, F. Provost, Learning when training data are costly: the effect of class distribution on tree induction, Journal of Artificial Intelligence Research 19 (2003) 315–354.

[51] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (3) (2011) 27.

[52] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27 (8) (2006) 861–874.

[53] K. Bache, M. Lichman, UCI machine learning repository, Tech. rep., University of California, Irvine (2013).

[54] M. Kubat, R. C. Holte, S. Matwin, Learning when negative examples abound, in: Proc. of ECML, 1997, pp. 146–153.