ORIGINAL RESEARCH

K Nearest Gaussian-A model fusion based framework for imbalanced classification with noisy dataset

Miao He[1], Jeffery D. Weir[2], Teresa Wu *[1, 3], Alvin Silva[4], Dianna-Yue Zhao[3], Wei Qian[3, 5]

[1] *School of Computing, Informatics, Decision Systems Engineering, Arizona State University, Tempe, USA*
[2] *Department of Operational Science, Graduate School of Engineering & Management, Air Force Institute of Technology, Write-Patterson AFB, USA*
[3] *Sino-Dutch Biomedical and Information Engineering School, Northeastern University, Shenyang, China*
[4] *Department of Radiology, Mayo Clinic, Scottsdale, USA*
[5] *Department of Electrical and Computer Engineering, University of Texas, El Paso, TX, USA*

Received: May 10, 2015 **Accepted:** July 28, 2015 **Online Published:** August 12, 2015
DOI: 10.5430/air.v4n2p126 **URL:** http://dx.doi.org/10.5430/air.v4n2p126

Abstract

Data quality issues such as data imbalance and data noise have great impact on the performances of many classifiers. Although the co-existence of imbalance and noise appears in many real world datasets, the issue of imbalance and noise have mostly been treated separately due to their different causes and problematic consequences. However, doing so may ignore the mutual effects thus may not achieve optimal classification performance. In this research, we propose a model fusion based framework, termed K Nearest Gaussian (KNG) to tackle the imbalance and noise issues jointly. KNG employs generative modeling method (GMM) to extract the data characteristics from the training data which are less sensitive to data imbalance and noise. The data characteristics are then used to establish Gaussian confidence regions which are used to achieve final classification in a K nearest neighbor (KNN) manner. Experiments on seven UCI benchmark datasets and one medical imaging dataset show KNG method greatly outperforms traditional classification methods in dealing with imbalanced classification problems with noisy dataset.

Key Words: Classification, Discriminative model, Generative model, K nearest neighbor, Gaussian mixture model

1 Introduction

Classification is a supervised learning problem which identifies the labels of new observations given a training dataset. Classification methods extract knowledge from the training dataset, and use the learned information to build models to predict the class of new observations. Therefore, the success of the classification methods highly depends on the quality of the training dataset. The real world datasets suffer from many quality issues.[1–3] Among them, the presences of imbalance and noise are the key factors which draw great attentions.[3–5] Data imbalance occurs when one class (minority class) is greatly outnumbered by another class (majority class). Most classification methods generally tend to underestimate the minority class due to the fact that majority class dominates the whole dataset. As a result, the performance of most classification methods degrades for imbalanced dataset. One special case of imbalanced classification is one-class classification (a.k.a. unary classification)

*Correspondence: Teresa Wu; Email: Teresa.Wu@asu.edu; Address: School of Computing, Informatics, Decision Systems Engineering, Arizona State University, Tempe, AZ, 85287-5906, USA.

where learning is conducted on the training data containing only the instances of one class. Of particular interest, this research is for more general imbalanced classification problem which focuses on learning from both minority class and majority class. Data noise occurs when the data has been corrupted by various reasons such as systematic uncertainty, measurement error, human error, *etc.*[2,5] It can be characterized as (1) attribute noise, which refers to the corruption in the features, and (2) class noise, which occurs when the instances are incorrectly labeled. Noise may hinder the knowledge extraction from the data and thus makes the classifier less effective, particularly if the classifier is noise-sensitive.

Data imbalance and data noise often co-exist in the real world datasets, that is, the dataset is imbalanced as well as noisy. Taking the CT imaging dataset as an example, the cancer patient often has a small portion of cancer tissues compared with normal tissues on the CT images which makes the dataset imbalanced. And the reconstruction method[6] used to generate the CT images comes with a systematic uncertainty making the images inherently noisy. Data imbalance affects the learning by degrading the recognition power of the classifier on the minority class because the majority class dominates, while data noise affects the learning by providing inaccurate information to the classifier and thus misleads the classifier. Because of these differences, data imbalance and data noise issues have been treated separately in the data mining field. Yet, such approaches ignore the mutual effects among imbalance and noise may lead to new problems. For example, data cleaning techniques[7] have been widely used in dealing with data noise which removes the noisy instances. If the removed instances happen to be the minority class, doing so may aggravate the level of imbalance. On the other hand, oversampling method such as synthetic minority oversampling technique (SMOTE),[8] which has been widely used for imbalanced datasets, may cause the data even noisier if the oversampled instances happen to be the noisy ones. One may argue that techniques may be carefully chosen to handle the data imbalance followed by data noise or vice versa, however, this two-step procedure may not be computational efficient. A desirable solution is to tackle these two issues jointly.

Most research on addressing the imbalance and noise employs discriminative models[9] which are effective in finding the class boundaries.[9,10] However, discriminative models are sensitive to data imbalance and noise though, since they work on the raw training data directly. Alternatively, generative models[9] study the probability distribution of the training data and extract data characteristics from the training data which can be used to achieve classification. This is also known as semi-supervised learning which is considered as an extension of classification with added probabilistic information. Generative models may be less effective in identifying the class boundaries than discriminative models.

Noticing the complementary nature of the generative and discriminative classifiers, in this research, we propose a novel generative-discriminative model fusion based framework, termed K Nearest Gaussian (KNG). A generative classifier, Gaussian Mixture Model (GMM) is used to model the training data as Gaussian mixtures and form adjustable confidence regions of each Gaussian. GMM is chosen here due to its capability in modeling arbitrary shaped densities.[11] Motivated by the idea of K-nearest neighbors (KNN), KNG finds nearest Gaussians to classify the testing data instances. To validate the performance of KNG, we use 7 UCI benchmark datasets. We purposely modify the datasets to make them imbalanced and noisy. The experimental study shows that KNG method is more effective and robust than other widely used classification methods, such as Support Vector Machine (SVM),[12] Artificial Neural Network (ANN),[13] Decision Tree (C4.5)[14] and KNN.[15] We further conduct a case study on a medical imaging dataset to test the applicability of KNG in real world application. The result also shows that KNG outperforms other commonly used classification methods.

2 Literature review

2.1 Review of techniques on handling imbalanced data

Presently, there are a number of studies attempting to overcome the classification problem with imbalance issue. They can be categorized into two approaches: data-level approach and algorithm-level approach.

The data-level approach uses different sampling techniques to increase/decrease the size of the training data in order to generate a balanced dataset. The representative methods are: undersampling,[4] oversampling[4] and SMOTE.[8] Undersampling randomly removes the data instances of majority class and thus may lead to information loss. Oversampling increases the size of the data by generating replicates of minority class. One possible way is to add Gaussian noise from the same distribution to the replicates to properly present the original dataset.[16] However, it is known oversampling may lead to over fitting.[3] SMOTE oversamples the minority class by generating artificial data which are the convex combination of the existing ones and thus improves learning. However, SMOTE may not perform well when the data instances used to generate new instances happen to be outliers and noisy examples.[17] Generally, the data-level approach alters the original training data distributions to make the dataset less imbalanced. However, the change of original data may compromise the underlying knowledge of the training data and thus is expected to be avoided.

The algorithm-level approach augments the existing methods to make them less sensitive to data imbalance. Many of

the existing studies tackle the imbalance data by developing extensions of SVM. For example, boundary movement (BM-SVM)[18] method changes the threshold value in SVM decision function to push the class boundary towards the majority class, Kernel-boundary alignment (KBA)[19] modifies the kernel matrix used in SVM training, and cost-SVM (cSVM)[20] applies different penalties to different classes. There are also a number of studies on extensions of ANN to tackle the imbalance issue. For example, two-step ANN[21] optimizes the weights and decision threshold values by using particle swarm optimization (PSO) to recognise the monirity class, HIPPO method[22] trains the ANN in a novelty detection approach, and cost sensitive ANN[23] integrates the misclassification cost to ANN. In summary, most of the algorithm-level approaches are extensions of the base classifiers such as SVM and ANN. Generally, these extensions are algorithm dependent and application dependent. Thus their effectiveness is limited by certain application context.

2.2 Review of techniques on handling noisy dataset

The existing noise handling techniques can also be categorized into two approaches: data-level approach and algorithm-level approach.

Data-level approach, also known as noise elimination techniques, handles the noise issue by removing the noise instances from the training data. For example, AJAX method[7] uses four types of data transformations—mapping, matching, clustering, and merging to detect and remove the noise data, Brodley and Friedl[24] compare the single algorithm filter, majority vote filter and consensus filter to identify and eliminate mislabeled training instances, Miranda *et al.*[25] combine the prediction of four different machine learning methods to guide the noise detection and removal. These data-level approaches focus on detecting and removing the noise instances. However, these methods generally cannot distinguish the noise cases from the rare cases. The removal of rare cases may lead bias to the training data. In addition, noise instances which contain error in some features may still contain useful information in other features. Thus, the removal of noise under this circumstances may lead to loss of valuable information.

Algorithm-level approach tackles the noisy dataset by improving the learning process of an algorithm to make it less sensitive to data noise. For example, Pechenizkiy *et al.*[26] use feature extraction technique as a preprocessing step in the training to diminish the effect of class noise, Mingers[27] compares different search heuristics and stopping criteria in decision tree construction in dealing with noise data, Quinlan[28] applies a post-pruning decision tree building procedure to deal with noise data. Although most of the algorithm-level approaches do not require data preprocessing, they are generally algorithm dependent or application dependent, thus are effective only when applied under certain context.

As a summary of both imbalance handling and noise handling techniques, data-level approach alters the original distribution of training dataset which may lead to loss of valuable information and thus is expected to be avoided. The algorithm-level approach is developed based on existing classifiers (such as SVM, ANN, C4.5), all of which employ discriminative models which are sensitive to data imbalance and noise since they work on the raw training data directly.

3 Proposed approach: K Nearest Gaussian

In this study, we propose a novel method, K Nearest Gaussian (KNG). Specifically, we employ a generative model, GMM, into the training process to extract the data characteristics from training data. GMM has been shown promising in dealing with data imbalance issue in our previous study[29] since the extracted data characteristics are expected to be less sensitive to data imbalance and noise. The idea of KNN finding the class boundary is adopted here to differentiate the classes based on the extracted data characteristics. In this section, we first review the basics of KNN in section 3.1 and GMM in section 3.2 followed by the details of our proposed KNG in section 3.3.

3.1 K nearest neighbor

KNN is a discriminative model that classifies instances based on the majority voting of its k nearest neighbors.[30] Figure 1 is the illustration example of KNN algorithm.

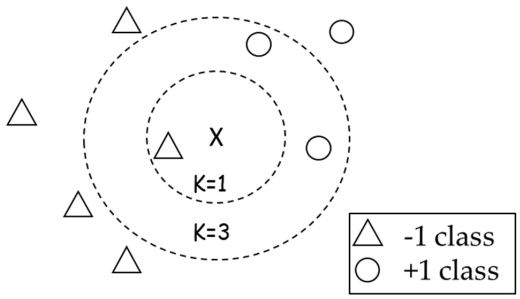

Figure 1: Illustration example of KNN algorithm

In Figure 1, X is a testing instance, circles and triangles are positive and negative class instances, respectively. KNN first calculates the distances from X to other training instances, and classify X according to the majority voting of its k nearest neighbors. K is predefined by the user. In Figure 1, when $k = 1, X$ is classified as negative class since the nearest neighbor is negative, while when $k = 3, X$ is classified as positive class since the majority of its three nearest neighbors is positive. Thus, X can be classified based on the neighboring instances.

3.2 Gaussian mixture model

GMM is a generative model which is widely used to model the distribution of the training data.[31,32] GMM is used in this research due to its well-known property in modeling arbitrary shaped densities without pre-assumptions on the distribution.[11] In addition, GMM has less parameters to tune compared with other generative models such as Hidden Markov model[33] or Restricted Boltzmann machine.[34] GMM models the probability density function of the feature vector x by using a mixture of weighted Gaussians as shown in equation 1:

$$P_{GMM}(x|y_i) = \sum_{m=1}^{M} c_{im} N(x, \mu_{im}, \sigma_{im}^2) \quad (1)$$

Where:

$$N(x, \mu_{im}, \sigma_{im}^2) = \frac{1}{(2\pi\sigma_{im}^2)^{\frac{d}{2}}} e^{-\frac{1}{2}\frac{\|x-\mu_{im}\|^2}{\sigma_{im}^2}} \quad (2)$$

C_{im}, μ_{im} and σ_{im}^2 are the weight, mean and covariance of the m^{th} mixture for class i. M is the number of mixtures which is predefined by the user. GMM method is an unsupervised method reflecting the intra-class information. Given a training dataset with binary class labels $\{(x_1, y_1), \cdots, (x_n, y_n)\}, y \in \{-1, 1\}$, the data are first divided into two groups by their class labels. Then the coefficients C_{im}, μ_{im} and σ_{im}^2 are computed using the Expectation Maximization (EM) algorithm[35] to find maximum likelihood function of the parameters iteratively.

3.3 K Nearest Gaussian (KNG)

Inspired by the KNN algorithm, which classifies an instance based on neighboring instances, we propose our KNG algorithm to tackle the imbalance and noise data issues. Instead of using the neighboring data instances, KNG uses the neighboring Gaussian mixtures to achieve classification. Specifically, KNG first applies GMM method to model the distributions of each class, and the data characteristics (such as centroid, variance) of each Gaussian can be then used to calculate the distances of the testing instance to the confidence region of each Gaussian. The smaller the distance, the higher probability that the testing instance belongs to the corresponding Gaussian distribution. Based on the distance to each Gaussian, the testing instance can be classified by majority voting. The data characteristics extracted by GMM method, comparing with raw training data, are expected to be less sensitive to imbalanced and noisy dataset. This makes KNG a promising method to deal with imbalanced and noisy data. The notations and pseudo code of KNG algorithm can be found in Table 1 and Figure 2.

Table 1: Notations used in KNG algorithm

Symbol	Meaning
X_{train}	Training dataset
X_{test}	Testing dataset
y	True label
y^{pred}	Predicted label
NumF	Number of folds in cross validation
n^+ , n^-	Number of Gaussian centers for +1/-1 class
μ^+, σ^{2+}	Centers and variances for GMM (+1 class)
μ^-, σ^{2-}	Centers and variances for GMM (-1 class)
β_+	Confidence region adjusting coefficient (+1 class)
β_-	Confidence region adjusting coefficient (-1 class)
A	Search range of β_1
B	Search range of β_2
k	Number of nearest Gaussians
CM	Confusion matrix
EvalMetric	Evaluation metric

```
Input:
        X_train ; /* training data */
        X_test ; /* testing data */
        K; /* number of nearest Gaussians */
        n+; /* number of Gaussian centers for positive class */
        n-; /* number of Gaussian centers for negative class */
        A; /* search range of β1 */
        B; /* search range of β2 */
Output:
        bestEvalMetric; /* the best Evaluation metric found */
        Classifier; /* output classifier with EvalMetric*/
Function Calls:
        GMMtrain (); /* train GMM classifier */
        ComputeDist_PR (); /* compute point to region distance */
        Sort (); /* sort the distances in ascending order */
        ComputeCM (); /* compute confusion matrix */
        ComputeEval (); /* compute evaluation metrics */
Begin
1) foreach β+ ∈ A
2)   foreach β- ∈ B
3)     for h= 1: NumF
4)       [μ+, σ2+, μ-, σ2-] ← GMMtrain (X_train^h, n+ , n-);
5)        foreach xi ∈ X_test^h
6)          foreach j ∈ n+
7)            Dist_PR (xi, j) ← ComputeDist_PR (xi, μ_j+, σ_j2+, β+);
8)          end foreach
9)          foreach q ∈ n-
10)           Dist_PR (xi, q + n+) ← ComputeDist_PR (xi, μ_q-, σ_q2-, β-);
11)         end foreach
12)          [order] ← Sort (Dist_PR(xi,:));
13)          y_i^pred = sum(y(order(1:K)));
14)       end foreach
15)     end for
16)     CM ← ComputeCM (y, y^pred);
17)     EvalMetric ← ComputeEval (CM);
18)     if EvalMetric >= bestEvalMetric
19)       then bestEvalMetric ← EvalMetric
20)     end if
21)   end foreach
22) end foreach
23) return [bestEvalMetric, Classifier];
End
```

Figure 2: Pseudo code for KNG Algorithm

In KNG algorithm, the *ComputeDist_PR* function is used to compute point to region distance, which is defined as fol-

lowing:

$$Dist\_PR(x_i, \mu_i, \sigma_i^2, \beta) = EuclideanDist(x_i, \mu_i) - \beta\sigma_i \quad (3)$$

β_+ and β_- are used to adjust the radius of the confidence region for positive(minority) and negative (majority) Gaussians, respectively. They can be seen as weights for positive/negative classes. The unequal settings of β_+ and β_- afford the KNG algorithm the flexibility to favor one class more than the other. This property is very useful in dealing with imbalanced data in which the majority class dominates. Thus, by assigning higher β_+, KNG can be more inclined to positive class and more positive instances can be recognized. This can be shown in the following illustration example. In Figure 3, we apply GMM to find the Gaussian mixtures for positive/negative classes. Circles are positive instances and triangles are negative instances. The Gaussian mixtures are represented by the concentric circles where different circles represent different β values.

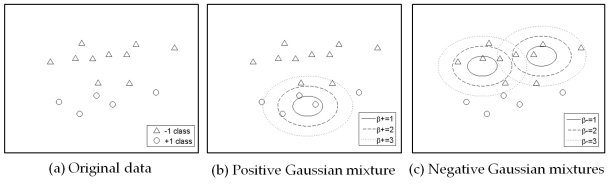

Figure 3: Finding Gaussian mixtures for positive/negative classes

KNG algorithm has five parameters to tune in order to achieve its best classification performance: number of nearest Gaussians k, number of positive Gaussians n^+, number of negative Gaussians n^-, and adjusting factors β_+, β_-. Number of nearest Gaussians k adjusts the number of Gaussians that will be used in finding the class boundary. When k is small, only the nearby Gaussians are used in finding the boundary, while when k is large, many far-away Gaussians are involved in finding the boundary.

Figure 4 shows the impact of the number of Gaussians to formation of class boundary. We keep k, β_+, β_-, n^+ as constant (all equal to one) while just change n- to see how the increase of number of Gaussians for one class would affect the formation of class boundary. When n^- equals n^+, the two classes are linearly separated by a straight line. When we increase n^- to 2 (see Figure 4b), the class boundary bends more towards the positive class (dark gray region) and thus more instances can be classified as negative. In addition, the linear boundary (see Figure 4a) becomes the intersection of two linear borderlines. If we further increase n^- (see Figure 4c), the class boundary can be further refined, which shows as two intersections of three linear borderlines. However, one potential issue with the increased number of Gaussians is overfitting. It is our intention to assess the robustness of the proposed KNG with the increasing number of Gaussian for both +/- classes.

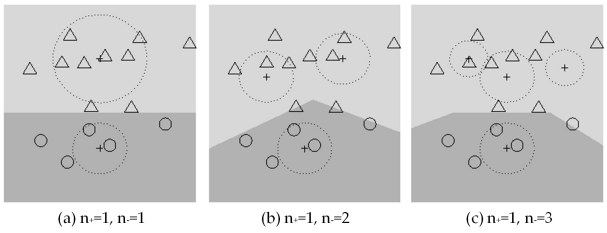

Figure 4: Impact of number of Gaussians to formation of class boundary

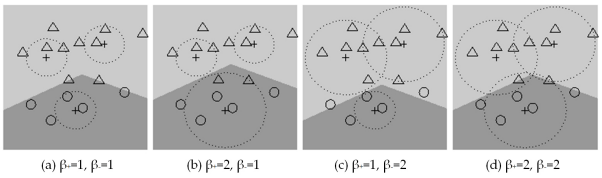

Figure 5: Impact of different β_+, β_- to formation of class boundary

Figure 5 shows different settings of β_+ and β_- can push the class boundary towards certain classes. To make it simple, we assume that all Gaussian mixtures have same variance. Figure 5a shows the positive (dark gray) and negative (light gray) class regions with the equal setting of β_+ and β_- (β_+=1, β_-=1). The border of the two regions is the class boundary. In Figure 5b and 5c, we observe that increasing $\beta_+(\beta_+ = 2, \beta_- = 1)$ can push the boundary towards negative class and thus more instances can be classified as positive while increasing $\beta_-(\beta_+ = 1, \beta_- = 2)$ can push the boundary towards positive class and thus more instances can be classified as negative. As aforementioned, β_+ and β_- are used as class-specific weights to adjust the radius of the confidence region for positive/ negative Gaussians (circles with dash lines). Thus the tuning of β_+ and β_- can push the class boundary towards certain class. For imbalanced datasets, the class boundary always skews towards the positive class since the negative class dominates. Thus, by assigning higher β_+, KNG can push the class boundary to positive class and more positive instances can be recognized.

4 Experiments and results

In this section, we first evaluate the performance of KNG using seven UCI benchmark datasets. Next, in a case study we use a medical imaging dataset to test KNG on real world application. To evaluate the performance of the classifier, we use Gmean measure which has been widely used[36–38] on imbalanced classifier for its ability to evaluate the performance of a classifier on both positive and negative classes. Gmean is defined as $\sqrt{acc^+ * acc^-}$, where acc^+

(also called sensitivity) and acc^- (also called specificity) are positive and negative class prediction accuracy, respectively. Area Under the Curve (AUC[4]) measure is also provided.

4.1 UCI benchmark datasets

The seven benchmark datasets we used in the experiments are collected from UCI Machine Learning Repository.[39] We call these datasets original datasets. The details of the original datasets are summarized in Table 2. The multiclass datasets are preprocessed as binary class problems, and the number in the name indicates the positive class. For example, in iris2, class 2 is used as positive class and all the other classes in the original data have been joined to represent the negative class. Based on the original datasets, we generate the imbalanced datasets by randomly removing 80% of the negative class instances. Then, we further add 20% of random noise to make the datasets both imbalanced and noisy. The noise is introduced using the following rules as literatures[5] did:

- Class noise: 20% of the class labels are randomly replaced by the opposite class labels
- Attribute noise: 20% of each attribute are replaced by random values from the domain (value range) of that attribute

Table 2: The UCI dataset used in the experiments

Dataset	#Instance	#Features	Imbalance Ratio of Original dataset	Imbalance Ratio of Imbalanced dataset
breast_ cancer	683	10	1.9	9.3
diabetes	768	8	1.9	9.3
iris2	150	4	2	10.0
mammo- graphic	830	5	1.1	5.3
yeast1	1484	8	2.2	11.0
wine2	178	13	1.5	7.6
glass3	214	9	1.8	9.2

We compare the performance of KNG method with SVM, ANN, C4.5 and KNN. These methods are chosen because they are widely used in classification problems. The KNG method is developed using MATLAB. SVM is performed using the libsvm MATLAB codes.[40] ANN, C4.5 and KNN are performed using a machine learning software WEKA 3.6.9.[41] In this study, we use grid search technique[42] in the parameter tuning process since it's easy to implement. The search ranges of the parameters are summarized in Table 3. Each method is performed using 10 fold cross validation. Because of the random nature of GMM method, the result of KNG algorithm is performed 20 times for each dataset, and the mean and standard deviation are reported.

Table 3: Search ranges of Parameters

Method	Parameter	Range
SVM(rbf_kernel)	γ	0-512
	C	0-2048
C4.5	confidence factor	0.1-0.5
KNN	# nearest neighbors k	1-9
ANN	learning rate	0.1-0.8
KNG	# nearest Gaussians k	1-5
	#centers(+1 class, -1 class)	1-5
	adjusting factors β_+, β_-	1-3

Table 4 shows the experimental results of Gmean measures for both original and imbalanced and noisy (I+N) datasets. For original datasets, KNG achieves best Gmean in three out of seven datasets, and for iris2, wine2 datasets, KNG is just marginal worse than the best method. This shows that KNG is comparable to other classification methods on original datasets. For I+N datasets, KNG greatly outperforms other methods in all seven datasets: for breast_cancer dataset, KNG (0.967) outperforms the second best method SVM (0.787) by 0.180; for diabetes dataset, KNG (0.708) outperforms the second best method ANN (0.331) by 0.377; for iris2 dataset, KNG (0.941) outperforms the second best method ANN (0.763) by 0.178; for mammographic dataset, KNG (0.789) outperforms the second best method KNN (0.564) by 0.225; for yeast1 dataset, KNG (0.624) outperforms the second best method KNN (0.418) by 0.206; for wine2 dataset, KNG (0.967) outperforms the second best method KNN (0.676) by 0.291; for glass3 dataset, KNG (0.721) outperforms the second best method SVM (0.509) by 0.212. In summary, the average outperformance of KNG to the second best method is 0.24. In all, KNG method is very effective in dealing with imbalanced classification problem with noisy datasets.

As shown in Table 5, the AUC measures are similar to Gmean. KNG does not show outperformance on the original dataset, but for I+N datasets, KNG outperforms all other methods in all seven datasets. For datasets such as mammographic and yeast1, methods such as ANN show less than 0.5 AUC measures which indicates worse than random performance. However, on these datasets, KNG shows much better AUC measures ($0.676\pm$ 0.000). We conclude KNG is effective in dealing with imbalanced and noisy data.

We further analyze the robustness of each method using the change of Gmean and the change of AUC which are defined as the measures on I+N datasets minus that of original datasets. These metrics show that to what extent the coexistence of imbalance and noise can affect the performance of a classifier. Small change of Gmean (or change of AUC) would indicate the model is robust since it is less affected by imbalance and noise. As seen, the performance of SVM, C4.5, ANN and KNN drop dramatically on I+N datasets compared with on original datasets. However, KNG main-

tains the minimal change of both Gmean and AUC for all seven datasets, which is shown in Table 6 and Table 7. The average change of Gmean for KNG is less than 1.6%, and change of AUC is 1.3%, both are significantly better than the other four methods. This is because the traditional classification methods, SVM, C4.5, ANN, KNN work on the training raw data directly which is sensitive to data imbalance and noise and thus their performances are highly affected by the co-existence of imbalance and noise. KNG, on the other hand, is designed to work on data characteristics extracted from the training data which are less sensitive to data imbalance and noise. As a result, KNG is able to preserve the performance when imbalance and noise co-exist in datasets. We conclude KNG is robust in dealing with both imbalance and noise issues.

Table 4: Gmean metric

Dataset	SVM		C4.5		ANN		KNN		KNG	
	Orig	I+N	Orig	I+N	Orig	I+N	Orig	I+N	Orig	I+N
breast_cancer	0.976	0.787	0.959	0.000	0.962	0.517	0.970	0.457	**0.976 ± 0.002**	**0.967 ± 0.000**
diabetes	0.712	0.136	0.690	0.000	0.710	0.331	0.683	0.283	**0.723 ± 0.002**	**0.708 ± 0.000**
iris2	0.954	0.548	0.910	0.000	**0.960**	0.763	**0.960**	0.000	0.957 ± 0.014	**0.941 ± 0.011**
mammographic	0.836	0.111	**0.838**	0.435	0.816	0.237	0.800	0.564	0.797 ± 0.000	**0.789 ± 0.000**
yeast1	0.618	0.179	0.658	0.000	0.643	0.000	0.647	0.418	**0.675 ± 0.000**	**0.624 ± 0.000**
wine2	**0.986**	0.463	0.952	0.000	0.979	0.497	0.964	0.676	0.972 ± 0.000	**0.967 ± 0.000**
glass3	0.716	0.509	0.710	0.246	0.673	0.392	**0.808**	0.448	0.728 ± 0.014	**0.721 ± 0.053**

Table 5: AUC metric

Dataset	SVM		C4.5		ANN		KNN		KNG	
	Orig	I+N	Orig	I+N	Orig	I+N	Orig	I+N	Orig	I+N
breast_cancer	0.978	0.808	0.969	0.496	0.988	0.630	**0.990**	0.522	0.979±0.000	**0.965±0.001**
diabetes	0.753	0.570	0.764	0.500	**0.812**	0.549	0.785	0.480	0.746±0.000	**0.729±0.000**
iris2	0.971	0.790	0.945	0.485	0.994	0.785	**0.996**	0.485	0.958±0.018	**0.950±0.004**
mammographic	0.853	0.555	0.871	0.578	**0.887**	0.487	0.851	0.597	0.820±0.000	**0.810±0.000**
yeast1	0.700	0.562	0.755	0.497	**0.775**	0.497	0.745	0.467	0.696±0.000	**0.676±0.000**
wine2	0.988	0.682	0.955	0.500	**0.998**	0.524	0.982	0.677	0.984±0.000	**0.973±0.000**
glass3	0.785	0.657	0.707	0.486	0.727	0.484	**0.827**	0.510	0.763±0.028	**0.762±0.051**

Table 6: Robustness evaluation (Change of Gmean)

Dataset	SVM	C4.5	ANN	KNN	KNG
breast_cancer	-18.9%	-95.9%	-44.5%	-51.3%	-0.9%
diabetes	-57.6%	-69.0%	-37.9%	-40.0%	-1.4%
iris2	-40.6%	-91.0%	-19.7%	-96.0%	-1.6%
Mammographic	-72.5%	-40.3%	-57.9%	-23.6%	-0.8%
yeast1	-43.9%	-65.8%	-64.3%	-22.9%	-5.1%
wine2	-52.3%	-95.2%	-48.2%	-28.8%	-0.5%
glass3	-20.7%	-46.4%	-28.1%	-36.0%	-0.7%
Average	-43.8%	-71.9%	-42.9%	-42.7%	**-1.6%**

Table 7: Robustness evaluation (Change of AUC)

Dataset	SVM	C4.5	ANN	KNN	KNG
breast_cancer	-17.0%	-47.3%	-35.8%	-46.8%	-1.4%
diabetes	-18.3%	-26.4%	-26.3%	-30.5%	-1.7%
iris2	-18.1%	-46.0%	-20.9%	-51.1%	-0.8%
Mammographic	-29.8%	-29.3%	-40.0%	-25.4%	-2.0%
yeast1	-13.8%	-25.8%	-27.8%	-27.8%	-2.0%
wine2	-30.6%	-45.5%	-47.4%	-30.5%	-1.1%
glass3	-12.8%	-22.1%	-24.3%	-31.7%	-0.1%
Average	-20.1%	-34.6%	-31.8%	-34.8%	**-1.3%**

To further evaluate the applicability of KNG, a case study is conducted on a medical imaging dataset which is collected from Mayo Clinic, Arizona. The comparison experiment is discussed in the next section.

4.2 Renal stone dataset

The case study is conducted on a renal stone dataset which is collected from Department of Radiology, Mayo Clinic Arizona. This dataset is a Dual Energy CT dataset with 65 instances and 18 features for each instance, as shown in Table 8. In the 65 instances, 9 of them are cystine stones, and the rest 56 are non-cystine stones. The objective of this case

study is to test if KNG can be used to effectively distinguish cystine stones from non-cystine stones with the presence of unneglectable level of imbalance and noise in the data. This renal stone dataset has an imbalance ratio of 6.2, and the noise comes from the systematic uncertainty of the reconstruction methods which are used to generate the CT images, as we mentioned in section 1.

The classification of cystine stones is important in clinical practice for the following reasons. Firstly, cystine stone is too dense to be broken up by extracorporeal shock wave lithotripsy (a commonly used treatment method), which is effective in breaking other types of stones, such as uric acid stones, calcium oxalate stones and struvite stones, *etc*. Cystine stones are usually treated using percutaneous nephron lithotripsy which is designed for removing dense stones. Thus, the diagnosis of cystine stone has a significant impact on following treatment. Secondly, cystine stone is usually caused by cystinuria, which is a genetic autosomal recessive metabolic disorder.[43] Thus, the diagnosis of cystine stone indicates that the patient needs to take additional genetic screening tests other than medical treatment.[44]

We compare the performance of KNG with other machine learning algorithms which has been widely used in renal stone classification studies,[45,46] such as SVM, C4.5, ANN, KNN, Random Forest (RF) and Naive Bayes.

The experiments are performed using Weka 3.6.9 with 5-fold cross validation technique applied. In addition to Gmean, we also report sensitivity, specificity and AUC which are commonly used evaluation metrics for medical diagnosis field.

Table 8: The Renal Stone dataset

Dataset	#Classes	#Examples	#Positive	#Negative	IR	#Features	Feature Description
RenalStone	2	65	9	56	6.2	18	11 energy level measures 1 effective atomic number 6 material density measures

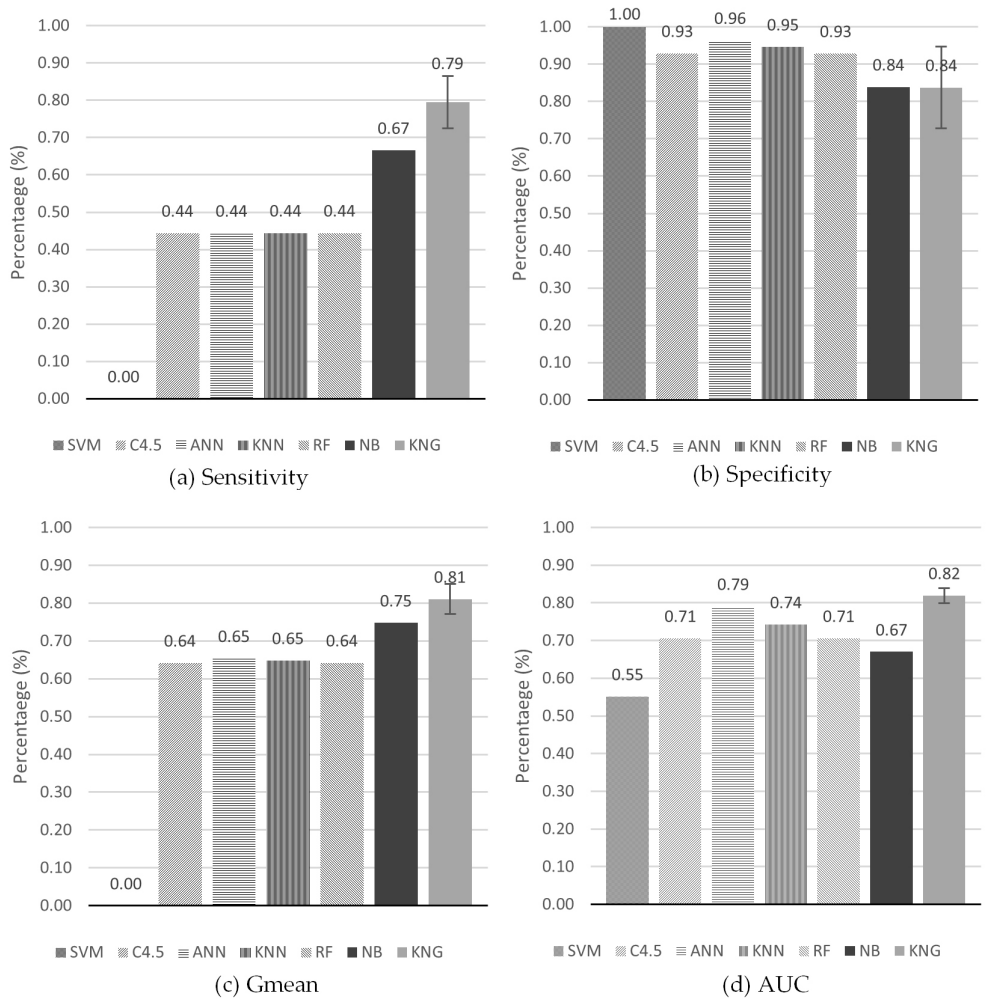

Figure 6: Sensitivity, specificity, Gmean and AUC metrics

From Figure 6, we see that SVM completely fails on this dataset. The zero sensitivity shows that SVM has no recognition ability of the cystine stones. C4.5, ANN, KNN and RF show similar performance with equal sensitivity (44%) and close specificity (93%, 96%, 95%, 93%, respectively), which lead to very similar Gmean measures (64%, 65%, 65%, 64%, respectively) and similar AUC measures(71%, 79%, 74%, 71%, respectively). NB shows equal specificity (84%) with KNG, but much lower sensitivity (67% *vs.* 79%), lower Gmean (75% *vs.* 81%) and lower AUC (67% *vs.* 82%). KNG achieves highest sensitivity (79%), highest Gmean (81%) and higest AUC (82%) among all seven methods while maintains high specificity (84%). In conclusion, KNG outperforms other six methods in classification of cystine stones.

5 Conclusion and discussion

In this research, we propose a discriminative and generative model fusion approach, KNG, to tackle classification problems with imbalance and noise issues jointly. Instead of modeling on the raw data directly, KNG applies GMM to model the training data as Gaussian mixtures and form adjustable confidence regions of each Gaussian which are less sensitive to data imbalance and noise. The classification is achieved by majority voting of the K nearest Gaussians for the testing instances. The experimental results on seven UCI datasets and one medical imaging dataset show that KNG is more effective in dealing with imbalanced dataset with noisy features than other commonly used classification methods.

In the experiments, we find the performance of KNG highly depends on the proper settings of parameters. As we can see in Table 3, there are five parameters to tune in the KNG algorithm, each of which has a wide search range. The parameters are tuned through grid search method in the experiments which is criticized to be computationally expensive and thus inefficient.[47] In addition, the search ranges of these parameters are determined by empirical experience which may not lead to optimal model performance. Facing all the above challenges, we plan to improve the KNG algorithm by employing advanced optimizer, such as Particle Swarm Optimization[48] to address the computation concerns as well as improving the parameter tuning performance as one future task. Secondly, one notable fact reported in this research is the experiments are conducted on mildly imbalanced (imbalanced ratio ranges from 5 to 11) dataset. We plan to further explore the applicability of KNG on heavily imbalanced problems (*e.g.*, 10e-5 in credit card fraud detection problem[16]) as another future work.

References

[1] Seiffert C, Khoshgoftaar TM, Hulse JV, *et al*. An empirical study of the classification performance of learners on imbalanced and noisy software quality data. Information Sciences. 2014; 259: 571-95. http://dx.doi.org/10.1016/j.ins.2010.12.016

[2] Zhu X, Wu X. Class noise *vs.* attribute noise: A quantitative study. Artificial Intelligence Review. 2004; 22(3): 177-210. http://dx.doi.org/10.1007/s10462-004-0751-8

[3] He H, Garcia EA. Learning from Imbalanced Data. Knowledge and Data Engineering. IEEE Transactions on. 2009; 21(9): 1263-84.

[4] Chawla NV. Data Mining for Imbalanced Datasets: An Overview. In: Data Mining and Knowledge Discovery Handbook, Springer; 2005. p. 853-67.

[5] Sáez JA, Galar M, Luengo J, *et al*. Tackling the Problem of Classification with Noisy Data using Multiple Classifier Systems: Analysis of the Performance and Robustness. Information Sciences. 2013; 247: 1-20. http://dx.doi.org/10.1016/j.ins.2013.06.002

[6] Hsieh J, Nett B, Yu Z, *et al*. Recent advances in CT image reconstruction. Current Radiology Reports. 2013; 1(1): 39-51. http://dx.doi.org/10.1007/s40134-012-0003-7

[7] Galhardas H, Florescu D, Shasha D, *et al*. AJAX: an extensible data cleaning tool. ACM SIGMOD Record. 2000; 29(2): 590. http://dx.doi.org/10.1145/335191.336568

[8] Chawla NV, Bowyer KW, Hall LO, *et al*. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research. 2002; 16: 321-57.

[9] Jordan A. On Discriminative *vs.* Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes. Advances in Neural Information Processing Systems. 2002; 14: 841.

[10] Lasserre J. Hybrid of generative and discriminative methods for machine learning, University of Cambridge; 2008.

[11] Lindsay BG. Mixture models: Theory, geometry, and applications. Mathematics. 1995.

[12] Cortes C, Vapnik V. Support-vector Networks. Machine Learning. 1995; 20(3): 273-97.

[13] Kriesel D. A brief introduction to neural networks. Retrieved August 15, 2011.

[14] Quinlan JR. C4. 5: programs for machine learning. Morgan kaufmann. 1993; 1.

[15] Tan PN, Steinbach M, Kumar V. Introduction to data mining. Pearson Addison Wesley; 2006.

[16] Salazar A, Safont G, Vergara L. Surrogate techniques for testing fraud detection algorithms in credit card operations. In: Security Technology (ICCST). International Carnahan Conference on. 2014. p.1-6.

[17] He H, Bai Y, Garcia EA, *et al*. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. In: Neural Networks, IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, 2008.

[18] Wu G, Chang EY. Class-boundary Alignment for Imbalanced Dataset Learning. In: ICML 2003 Workshop on Learning from Imbalanced Data Sets II, Washington, DC; 2003.

[19] Wu G, Chang EY. Aligning Boundary in Kernel Space for Learning Imbalanced Dataset. In: Data Mining, ICDM'04. Fourth IEEE International Conference on. IEEE, 2004.

[20] Veropoulos K, Campbell C, Cristianini N. Controlling the Sensitivity of Support Vector Machines. In: Proceedings of the International Joint Conference on Artificial Intelligence; 1999.

[21] Adam A, Ibrahim Z, *et al*. A Two-Step Supervised Learning Artificial NeuralL Network for Imbalanced Dataset Problems. Inter-

national Journal of Innovative Computing Information and Control. 2012; 8(5a): 3163-72.

[22] Japkowicz N, Myers C, Gluck M. A novelty detection approach to classification. In: IJCAI; 1995.

[23] Berardi VL, Zhang GP. The effect of misclassification costs on neural network classifiers. Decision Sciences. 1999; 30(3): 659-82.

[24] Brodley CE, Friedl MA. Identifying mislabeled training data. arXiv preprint, no. arXiv:1106.0219, 2011.

[25] Miranda AL, Garcia LPF, Carvalho AC, *et al*. Use of classification algorithms in noise detection and elimination. In: Hybrid Artificial Intelligence Systems; 2009. p. 417-24.

[26] Pechenizkiy M, Tsymbal A, Puuronen S, *et al*. Class noise and supervised learning in medical domains: The effect of feature extraction. In: Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on 2006.

[27] Mingers J. An empirical comparison of selection measures for decision-tree induction. Machine learning. 1989; 3(4): 319-42.

[28] Quinlan JR. The effect of noise on concept learning. In: Machine learning: An artificial intelligence approach, Morgan Kaufmann; 1986. p. 149-66.

[29] He M, Wu T, Silva A, *et al*. Augmenting Cost-SVM with Gaussian Mixture Models for Imbalanced Classification. Artificial Intelligence Research. 2015; 4(2): 93.

[30] Cover T, Hart P. Nearest neighbor pattern classification. Information Theory, IEEE Transactions on. 1967; 13(1): 21-7.

[31] Wang K, Ren Z. Enhanced Gaussian Mixture Models for Object Recognition using Salient Image Features. In: International Conference on Mechatronics and Automation, ICMA; 2007.

[32] Reynolds DA, Rose RC. Robust text-independent speaker identification using Gaussian mixture speaker models. Speech and Audio Processing, IEEE Transactions on. 1995; 3(1): 72-83.

[33] Rabiner L, Juang BH. An introduction to hidden Markov models. ASSP Magazine, IEEE. 1986; 3(1): 4-16.

[34] Fischer A, Igel C. An introduction to restricted Boltzmann machines. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. 2012: 14-36.

[35] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal statistical Society. 1977; 39(1): 1-38.

[36] Akbani R, Kwek S, Japkowicz N. Applying Support Vector Machines to Imbalanced Datasets. In: Machine Learning: ECML 2004, Berlin Heidelberg, 2004.

[37] Wang HY. Combination Approach of SMOTE and Biased-SVM for Imbalanced Datasets. In: Neural Networks; 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on IEEE; 2008.

[38] Imam T, Ting KM, Kamruzzaman J. z-SVM: An SVM for Improved Classification of Imbalanced Data. In: AI 2006: Advances in Artificial Intelligence, Berlin Heidelberg, 2006.

[39] Bache K, Lichman M. UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science.

[40] Chang CC, Lin CJ. LIBSVM : A Library for Support Vector Machines. ACM Transac-tions on Intelligent Systems and Technology. 2011; 2(27): 1-27.

[41] Hall M, Frank E, Holmes G, *et al*. The WEKA Data Mining Software: An Update. SIGKDD Explorations. 2009; 11(1).

[42] Bergstra J, Bengio Y. Random search for hyper-parameter optimization. The Journal of Machine Learning Research. 2012; 13: 281-305.

[43] Wu J. Chapter 58 – Urolithiasis. In: Integrative Medicine, 3rd ed, WB Saunders Company, 2012.

[44] Breuning MH, Hamdy NA. From gene to disease; SLC3A1, SLC7A9 and cystinuria. Nederlands tijdschrift voor geneeskunde. 2003; 147(6): 245.

[45] Krishna Apparao R. Statistical and Data Mining Aspects on Kidney Stones: A Systematic Review and Meta-analysis. Open Access Scientific Reports. 2012.

[46] Kumar K, *et al*. Artificial Neural Networks for Diagnosis of Kidney Stones Disease. I.J. Information Technology and Computer Science. 2012; 7: 20-25.

[47] Bergstra J, Bengio Y. Random search for hyper-parameter optimization. The Journal of Machine Learning Research. 2012; 13: 281-305.

[48] Kennedy J. Particle swarm optimization. In: Encyclopedia of Machine Learning, Springer US; 2010. p. 760-6.