A Quadratic Mean based Supervised Learning Model for Managing Data Skewness

Wei Liu* Sanjay Chawla*

Abstract

In this paper, we study the problem of data skewness. A data set is skewed/imbalanced if its dependent variable is asymmetrically distributed. Dealing with skewed data sets has been identified as one of the ten most challenging problems in data mining research.

We address the problem of class skewness for supervised learning models which are based on optimizing a regularized empirical risk function. These include both classification and regression models for discrete and continuous dependent variables. Classical empirical risk minimization is akin to minimizing the arithmetic mean of prediction errors, in which approach the induction process is biased towards the majority class for skewed data. To overcome this drawback, we propose a quadratic mean based learning framework (`QMLearn`) that is robust and insensitive to class skewness. We will note that minimizing the quadratic mean is a convex optimization problem and hence can be efficiently solved for large and high dimensional data. Comprehensive experiments demonstrate that the `QMLearn` model significantly outperforms existing statistical learners including logistic regression, support vector machines, linear regression, support vector regression and quantile regression etc.

Keywords: Data skewness; quadratic mean; convex optimization.

1 Introduction

Learning from skewed/imbalanced data sets is a common yet challenging problem in supervised learning [6]. It has been identified as one of the ten most challenging problems in data mining research [33]. In this paper, we call a data set "skewed" if its dependent variable is *numerical* and asymmetrically distributed (Figure 1 shows an example); and call a data set "imbalanced" if its class variable is *categorical* and the number of instances in one class is different from those in the other class (we only consider two-class data sets in this study).

*Pattern and Data Mining Group, School of Information Technologies, the University of Sydney, Sydney NSW 2006, Australia. {wei.liu, sanjay.chawla}@sydney.edu.au

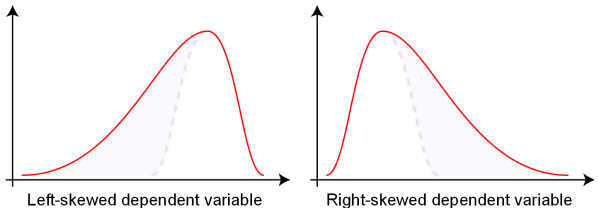


Figure 1: Examples of data sets with skewed numerical dependent variables.

In this paper we use "dependent variable" and "class" interchangeably even if the dependent variable is numeric.

The problem of handling imbalanced data sets from the perspective of classification has been well addressed in the knowledge discovery literature, while relatively less work has be done for dealing with skewed data for regression estimation. A common method for reducing the effects of data skewness on regression is to treat "long tail" data samples (samples far from the mean) as outliers and use quantile regression to estimate the median (instead of the mean) as the predicted value [24, 34].

In the literature of solving class imbalance problems, data-oriented methods use various sampling techniques to over-sample instances in the minor class or under-sample those in the major class, so that the resulting data is balanced. A typical example is the SMOTE method [5] which increases the number of minor class instances by creating synthetic samples. A variation [7] on SMOTE has integrated boosting with sampling strategies to better model the minority class, by focusing on difficult samples that belong to both minority and majority classes. It is recently proposed that using different weight degrees on the synthetic samples (so-called safe-level-SMOTE [3]) produces better accuracy than SMOTE.

The focus of algorithm-oriented methods has been put on the extension and modification of existing classification algorithms so that they can be more effective in dealing with imbalanced data. For example, an improved associative classifier named SPARCCC [29] has

been introduced to overcome the drawbacks of CBA [22] on imbalanced data sets. Besides, there are also modifications of decision tree algorithms on improving the standard C4.5, such as HDDT [8] and CCPDT [23].

Akbani et al. [1] and Tezel et al. [21] have proposed approaches to improve support vector machines (SVM) on imbalanced data sets, which they call SDC and GP respectively. But their work are both focused on improving sampling techniques (e.g. modifying SMOTE in GP) for SVM, and do not solve the problem of training bias in the design of SVM learning algorithm per se.

In this paper our goal is to improve the induction process of existing statistical learning algorithms. The model we propose is an algorithm-oriented method and we preserve all original information/distribution of the training data sets. This model is also a generic supervised learning framework that can be applied to the problem of regression estimation. More specifically, **the contributions** of this paper are as follows:

1. We express the traditional definition of empirical risk function as an arithmetic mean of prediction errors, from which perspective we illustrate why many existing statistical learners have undesirable performance on skewed or imbalanced data sets;
2. We redefine the empirical risk as a quadratic mean of scalar losses on each class label, which we call `QMLearn` method, making the regularized empirical risk functions robust and insensitive to class skewness / imbalance;
3. We apply the generic framework of `QMLearn` method to concrete learning models of logistic regression, linear SVM, ordinal linear regression, linear support vector regression (SVR) and quantile regression, and perform comprehensive empirical evaluations on data sets from publicly accessible data repositories, which testify to the superiority of our approach.

The rest of the paper is structured as follows. In Section 2 we introduce the traditional inductive principle of empirical risk minimization, and explain why it is flawed in learning from skewed/imbalanced data. We define `QMLearn` method in Section 3 and solve it as a convex optimization problem in Section 4. Experiments and analysis are reported in Section 5. We conclude in Section 6 with directions for future work.

2 Traditional Learning Models

Given training data $(\mathbf{x}_i, y_i)$ $(i = 1, ..., \text{n})$, where $\mathbf{x}_i \in \mathbb{R}^d$ are feature vectors, d is the number of features and y_i are class values ($y_i \in \{-1, 1\}$ for classification or $y_i \in \mathbb{R}$ for regression), many machine learning algorithms build predictive models through minimizing a regularized[1] risk function:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \ \lambda \mathbf{w}^T\mathbf{w} + R_{emp}(\mathbf{w}) \tag{2.1}$$

where $\mathbf{w}$ is the weight vector including the bias b (and hence $\mathbf{x}$ by default has an added feature $x_{d+1} \equiv 1$), λ is a positive parameter to balance the two items in Equation 2.1. $R_{emp}(\mathbf{w})$ is the empirical risk function which is in the form of:

$$R_{emp}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} l(x_i, y_i, \mathbf{w}) \tag{2.2}$$

where n is the total number of instances. The loss function $l(x_i,y_i,\mathbf{w})$ measures the discrepancy between a true label y_i and a predicted value from using $\mathbf{w}$. Different settings of loss functions yield different types of learners, such as $\ln(1 + \mathrm{e}^{-y\mathbf{w}^T\mathbf{x}})$ in logistic regression [9] and $\max(0, 1 - y\mathbf{w}^T\mathbf{x})$ as the binary hinge loss in SVM [13]. Here we use $\mathbf{w}^T\mathbf{x}$ to denote the dot product of a weight vector $\mathbf{w}$ and a feature vector $\mathbf{x}$. One important requirement for valid loss functions is that they must be convex and sub-differentiable, but not necessarily differentiable, as in the hinge loss function. Table 1 lists the loss functions of the learning models we use in this paper.

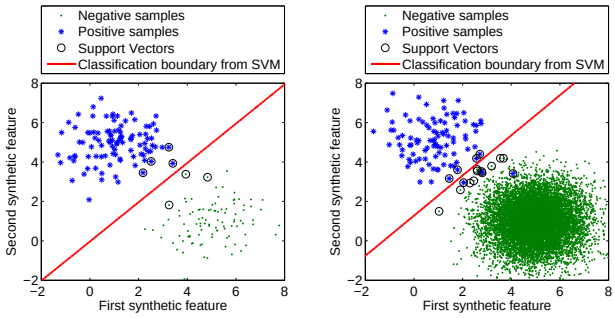
Now we explain the drawback of traditional statistical learners from the perspective of classification. For any form of loss functions, the discrepancy/loss between a true label and a predicted label occurs whenever there is a classification error. By minimizing the *arithmetic mean* (Equation 2.2) of such classification errors/discrepancies, one wants to solve the problem of Equation 2.1, and build decision boundaries from the final weight vector $\mathbf{w}^*$.

Figure 2 and 3 show the decision boundaries built by traditional logistic regression and linear SVM under different data distributions. We use two-dimensional feature space in the examples of the two figures, where instances are sampled from normal distributions with mean vector (5, 1) for the positive class and (1, 5) for the negative class, and the standard deviations of both dimensions on both classes are 1. We treat samples in the minor class as positive instances. Traditional logistic regression and linear SVM perform well when data is balanced, but when data is highly imbalanced we can observe that their decision boundaries are shifted towards the positive samples, which effectively minimizes the error rate only on negatives. On the imbalance data

[1] Without losing generality, in the rest of this paper we only consider the *L-2 norm* of the regularizer.

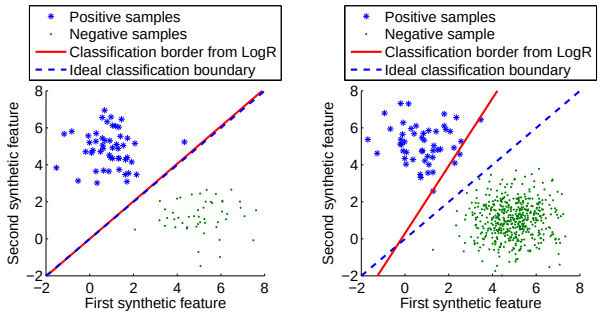
Table 1: Scalar loss functions and their derivatives.

	Scalar loss $l(\mathrm{x}_i,\mathrm{y}_i,\mathbf{w})$ (i=1,2,3...n)	Derivative $l'(x_i, y_i, \mathbf{w})$
Logistic Regression [9]	$\ln(\ 1 + \exp(\ -\mathrm{y}_i(\mathbf{w}^T\mathrm{x}_i))\ )$	$-\mathrm{y}_i/(1 + \exp(\ -\mathrm{y}_i(\mathbf{w}^T\mathrm{x}_i)))$
SVM with Hinge Loss [13]	$\max(0,\ 1-\mathrm{y}_i(\mathbf{w}^T\mathrm{x}_i))$	$-\mathrm{y}_i$ if $\mathrm{y}_i(\mathbf{w}^T\mathrm{x}_i) < 1$; otherwise 0
SVM with Huber Loss [4]	$\begin{cases} 1 - y_i(\mathbf{w}^T x_i) & \text{if } \ 1 - y_i(\mathbf{w}^T x_i) > h \\ \frac{(1+h-y_i(\mathbf{w}^T x_i))^2}{4h} & \text{if } \mid 1 - y_i(\mathbf{w}^T x_i) \mid \leq h \\ 0 & \text{otherwise} \end{cases}$	$\begin{cases} -y_i & \text{if } \ 1 - y_i(\mathbf{w}^T x_i) > h \\ \frac{-y_i(1+h-y_i(\mathbf{w}^T x_i))}{2h} & \text{if } \mid 1 - y_i(\mathbf{w}^T x_i) \mid \leq h \\ 0 & \text{otherwise} \end{cases}$
Ordinary Linear Regression [30]	$\frac{1}{2}(\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i)^2$	$\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i$
Support Vector Regression [28]	$\max(0,\ \lvert\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i\rvert - \epsilon)$	$\mathrm{sign}(\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i)$ if $\lvert\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i\rvert > \epsilon$; otherwise 0
Quantile Regression [20]	$\max(\tau(\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i),\ (1-\tau)(\mathrm{y}_i - \mathbf{w}^T\mathrm{x}_i))$	τ if $\mathbf{w}^T\mathrm{x}_i > \mathrm{y}_i$; otherwise $\tau - 1$



(a) Class ratio = 1:1 (b) Class ratio = 1:100

Figure 2: Performance of soft-margin SVMs on different class distributions. When data is imbalanced, the margin (classification boundary) in subfig. (b) maximizes the accuracy of the major class, and all misclassifications are from the minor class.



(a) Class ratio = 1:1 (b) Class ratio = 1:10

Figure 3: Performance of logistic regression on different class distributions. "LogR"in the legends represents traditional logistic regression. When data is imbalanced, traditional logistic regression minimizes the error rate well on the major class, but misclassifies many instances in the minor class (subfig. b).

settings of Figure 2(b) and 3(b), all misclassifications are made on positive samples.

As the goal of a classification task is to maximize both the sensitivity (i.e. $\frac{\#\text{ true positives}}{\#\text{ actual positives}}$) and the specificity (i.e. $\frac{\#\text{ true negatives}}{\#\text{ actual negatives}}$) simultaneously, we examine the relationship between these two measures and the arithmetic mean based error rates in terms of isometric lines shown in Figure 4. We can see that when data is balanced, the empirical risk is minimized when both sensitivity and specificity are close to 1. However, when data is imbalanced, one can obtain low empirical risks with high specificity but *low* sensitivity. This observation coincides with the scenario where conventional learners perform badly on positive samples in Figure 2(b) and 3(b).

3 Quadratic Mean based Learning Models

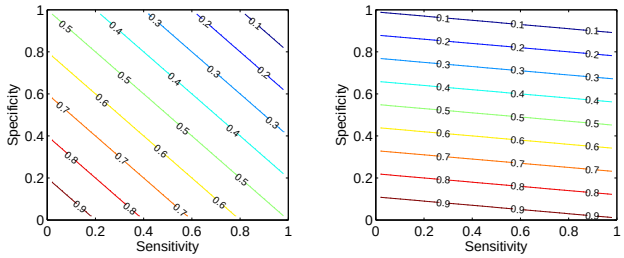
Quadratic mean (aka. "root mean square") measures the magnitude of varying quantities, which is defined as the square root of the arithmetic mean of the squares of each element to be averaged. For a set of values $\{x_1, x_2, ..., x_n\}$, its quadratic mean is:

$$\widetilde{x} = \sqrt{\frac{x_1^2 + x_2^2 + ... + x_n^2}{n}}$$

Denote by x_1 the classification error rate on one class and by x_2 that of the other class. We provide the following lemma and show that the quadratic mean of the two variables ($\sqrt{\frac{x_1^2+x_2^2}{2}}$) is lower bounded by their arithmetic mean ($\frac{x_1+x_2}{2}$), and this lower bound is reached if and only if $x_1 = x_2$.

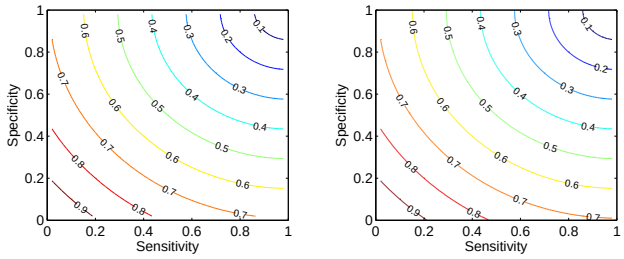
LEMMA 3.1. *$\forall\ x_1 \geq 0$, and $\forall\ x_2 \geq 0$, denote by AM and QM the arithmetic mean and the quadratic mean of x_1 and x_2 respectively, then $QM = \sqrt{AM^2 + (\frac{x_1-x_2}{2})^2}$.*

Proof. The lemma can be proved by the derivations of



(a) Original empirical risk on balanced data (b) Original empirical risk on imbalanced data (Pos:Neg = 1:10)

Figure 4: Loss measured by original empirical risk when data sets follow different class distributions. For balanced data, the empirical risk is minimized when both sensitivity and specificity are close to 1. But when data is imbalanced, the traditional empirical risk is insensitive to large changes of sensitivity.



(a) `QMLearn`-based empirical risk on balanced data (b) `QMLearn`-based empirical risk on imbalanced data (Pos:Neg = 1:10)

Figure 5: Loss measured by `QMLearn`-based empirical risk on different class distributions. No shifts of loss contour lines are observed when data becomes imbalanced.

the following equation:

$$\begin{aligned} QM &= \sqrt{\frac{x_1^2+x_2^2}{2}} \\ &= \sqrt{\frac{x_1^2+x_2^2+2x_1x_2}{4}+\frac{x_1^2+x_2^2-2x_1x_2}{4}} \\ &= \sqrt{(\frac{x_1+x_2}{2})^2+(\frac{x_1-x_2}{2})^2} \\ &= \sqrt{AM^2+(\frac{x_1-x_2}{2})^2} \qquad \square \end{aligned}$$

This lemma can also be intuitively explained by the Pythagorean theorem shown in Figure 6. We know that

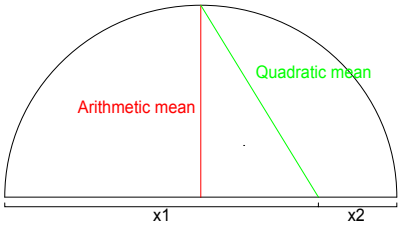


Figure 6: A geometric illustration of the arithmetic mean and the quadratic mean of x_1 and x_2 where $x_1 > x_2$. While the arithmetic mean is determined only by the sum of x_1 and x_2, the quadratic mean reflects both the sum and the difference of the two variables.

the arithmetic mean is minimized when the sum of x_1 and x_2 is minimal. Then from Lemma 1 it is clear that the quadratic mean is minimized only if the sum and the difference of x_1 and x_2 are both minimal.

EXAMPLE 1. *Consider a classifier with the following classification performance on an imbalanced data set:*

	Predicted Pos	*Predicted Neg*	*Sum*
Actual Pos	*1*	*9*	*10*
Actual Neg	*0*	*990*	*990*

Then the arithmetic mean of prediction errors on the two classes is $\frac{9+0}{10+990}$*=0.9% (by applying Equation 2.2), while its quadratic mean based error is* $\sqrt{\frac{(9/10)^2+(0/990)^2}{2}}$*=63.6%. So this (indeed biased) classifier has a low error rate from the arithmetic mean, but has a very high error rate estimated by the quadratic mean.*

Recall that traditional methods minimize the arithmetic mean of prediction errors in training process. From Example 1 we know that this minimization becomes more difficult to achieve if one uses the quadratic mean – it forces the minimization of not only the sum of prediction errors, but also the difference of errors between each class.

By using the notion of the quadratic mean, we redefine the empirical risk function of Equation 2.2 as:

(3.3)
$$R^Q_{emp}(\mathbf{w}) = \sqrt{\frac{(\frac{\sum_{i=1}^{\#pos} l(x_i,y_i,\mathbf{w})}{\#pos})^2+(\frac{\sum_{i=\#pos+1}^{n} l(x_i,y_i,\mathbf{w})}{\#neg})^2}{2}}$$

The notations in Equation 3.3 are under the assumption that positive instances are ahead of negative instances in a data set (i.e. $i \in (1,2,3,...,\#pos)$ are indexes of positive instances and the rest are those of the negatives).

We call Equation 3.3 the **`QMLearn`-based empirical risk function**, and call the solution of Equation 2.1 embedded with Equation 3.3 the **`QMLearn` method**.

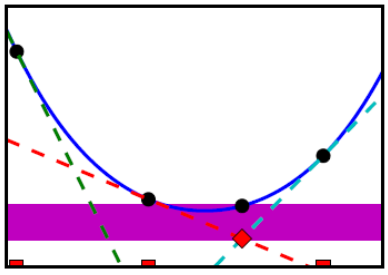


Figure 7: An illustration of the piecewise linear lower bound used in bundle methods [26]. In the figure is a convex function (i.e. $\mathrm{R}^{Q}_{emp}(\mathbf{w})$) with three subgradients evaluated at three different locations. The approximation gap at the end of third iteration is the difference between the lowest value of $\mathrm{R}^{Q}_{emp}(\mathbf{w})$ evaluated so far (lowest black circle) and the minimum of the lower bound (the red diamond).

To examine the robustness the definition of $R^{Q}_{emp}(\mathbf{w})$ against data imbalance, we also plot its relationship with sensitivity and specificity shown in Figure 5. It can be seen that the `QMLearn`-based empirical risk is always minimized when both sensitivity and specificity are close to 1, regardless of the distribution of the class variable.

3.1 `QMLearn` in Regression Estimation While Equation 3.3 is for the problem of classification, we now derived the `QMLearn`-based empirical risk function for regression tasks. Suppose instances in a numerical-class data set are sorted by the value of their dependent variable. We define the following empirical risk function to minimize the loss on both sides of the "median" of its dependent variable:

(3.4)
$$R^{Q}_{emp}(\mathbf{w}) = \sqrt{\frac{(\frac{\sum_{i=1}^{\frac{n}{2}} l(x_i,y_i,\mathbf{w})}{\frac{n}{2}})^2 + (\frac{\sum_{i=\frac{n}{2}+1}^{n} l(x_i,y_i,\mathbf{w})}{\frac{n}{2}})^2}{2}}$$

The rationale behind minimizing the prediction error rates on both side of the median is to eliminate the influences of left- or right-skewed data distributions (shown in Figure 1). In Section 5.3 we show that `QMLearn`-based regression works significantly better than pure median predictions of a 50% quantile regression.

3.2 Feature Selection The weights learned from Equation 2.1 have been used to suggest the significance of features in training data [16]. For example, in classical logistic regression, one wants to obtain the optimal vector of weights from solving:

Algorithm 1 Bundle methods for solving Equation 4.7

Input: convergence threshold ϵ; initial weight vector $\mathbf{w}_0$;
Output: minimizer $\mathbf{w}^*$ of Equation 4.7

Initialize iteration index $t \leftarrow 0$;
repeat
  $t \leftarrow t + 1$;
  Compute subgradient $a_t \leftarrow \partial_{\mathbf{w}} \mathrm{R}^{Q}_{emp}(\mathbf{w}_{t-1})$;
  Compute bias $b_t \leftarrow \mathrm{R}^{Q}_{emp}(\mathbf{w}_{t-1}) - \mathbf{w}^{T}_{t-1} a_t$;
  Update the lower bound $\mathrm{R}^{lb}_{t}(\mathbf{w}) := \max_{1\leq i\leq t} \mathbf{w}^T a_i + \mathrm{b}_i$;
  $\mathbf{w}_t \leftarrow \arg\min_w J_t(\mathbf{w}) := \lambda \mathbf{w}^T \mathbf{w} + \mathrm{R}^{lb}_{t}(\mathbf{w})$;
  Compute current gap $\epsilon_t \leftarrow \min_{0\leq i\leq t} J(\mathbf{w}_i) - J_t(\mathbf{w}_t)$
until $\epsilon_t \leq \epsilon$
Return $\mathbf{w}_t$

(3.5)
$$\min_{\mathbf{w}} \ \lambda \mathbf{w}^T \mathbf{w} + \frac{1}{n} \sum_{i=1}^{n} \ln(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i})$$

Since each feature weight (aka. "regression coefficient") describes to what extent that feature contributes to the class of interest, the features whose weights have large absolute values are considered important, and the ones with weights close to zero are considered trivial. When using `QMLearn` method to select features, we also use the absolute values of weights as indicators of feature significance, but different to Equation 3.5 we obtain the feature weights via solving:

(3.6)
$$\min_{\mathbf{w}} \ \lambda \mathbf{w}^T \mathbf{w} + \sqrt{\frac{(\frac{\sum_{i=1}^{\sharp pos} \ln(1+e^{-y_i \mathbf{w}^T \mathbf{x}_i})}{\sharp pos})^2 + (\frac{\sum_{i=\sharp pos+1}^{n} \ln(1+e^{-y_i \mathbf{w}^T \mathbf{x}_i})}{\sharp neg})^2}{2}}$$

It is also common to evaluate the worth of an attribute by using weights trained by linear SVM [14]. The differences between weights obtained from existing learners (e.g. Equation 3.5) and those from `QMLearn` methods (e.g. Equation 3.6) are reported in Section 5.4.

4 Convex Optimization

By substituting $R^{Q}_{emp}(\mathbf{w})$ for $R_{emp}(\mathbf{w})$ in Equation 2.1, we obtain the following `QMLearn` optimization problem:

(4.7)
$$\min_{\mathbf{w}} \ J(\mathbf{w}) := \lambda \mathbf{w}^T \mathbf{w} + R^{Q}_{emp}(\mathbf{w})$$

Calculating `QMLearn`-based empirical risk means taking the square root of the average of squares of originally convex scalar loss functions (shown in Table 1). Since the operation of taking squares preserves convexity, and the square root does not change the monotonicity of a function, the resulting quadratic mean of convex

Table 2: Data sets with categorical classes and classification comparisons among different classifiers. "#Inst" and "#Attr" denote the number of instances and attributes respectively. "Min(%)" represents the proportion of instances in the minority class. The numbers in brackets represent rankings of classifiers. The first row of Friedman tests compare conventional learners with `QMLearn` learners, and the second row of tests compare the best `QMLearn` learner (i.e. "LogR^Q") with all the other learners.

Name	#Inst	#Attr	Min(%)	CV	Area Under Precision-Recall Curve											
					LogR	LogR^Q	Hinge	Hinge^Q	Huber	Huber^Q	CCPDT	HDDT	SDC	GP	Liblinear	ASVM
KDDCup'09[7]:																
Appetency	50000	231	1.8	46503	.023(9)	.058(1)	.024(6)	.026(5)	.024(6)	.027(4)	.018(11)	.019(10)	.046(2)	.042(3)	.012(12)	.024(6)
Churn	50000	231	7.3	36390	.079(10)	.123(1)	.081(8)	.093(6)	.081(8)	.101(5)	.084(7)	.078(12)	.109(3)	.118(2)	.079(10)	.109(3)
Upselling	50000	231	7.4	36356	.087(10)	.309(1)	.086(11)	.142(9)	.086(11)	.146(8)	.246(6)	.238(7)	.293(4)	.304(2)	.265(5)	.297(3)
UCI [2]:																
Anneal	898	39	4.5	745	.048(12)	.478(7)	.050(10)	.765(3)	.049(11)	.763(4)	.882(2)	.918(1)	.359(8)	.530(6)	.122(9)	.579(5)
Arrhythmia	452	280	4.9	368	.055(6)	.052(9)	.055(6)	.043(11)	.055(6)	.043(11)	.056(5)	.051(10)	.061(3)	.062(2)	.059(4)	.071(1)
Cylinder	540	40	42.2	13	.688(7)	.735(1)	.697(5)	.718(2)	.697(5)	.718(2)	.588(10)	.466(12)	.673(8)	.705(4)	.579(11)	.672(9)
German	1000	21	30.0	160	.554(7)	.613(3)	.550(9)	.624(1)	.551(8)	.623(2)	.368(12)	.435(10)	.555(6)	.605(4)	.423(11)	.597(5)
Hypothyroid	3772	30	2.5	3401	.031(10)	.737(5)	.031(10)	.859(4)	.031(10)	.864(3)	.901(1)	.893(2)	.386(9)	.723(6)	.655(8)	.707(7)
Ionosphere	351	35	35.9	27	.825(6)	.849(3)	.866(1)	.821(7)	.866(1)	.821(7)	.793(11)	.811(10)	.830(5)	.832(4)	.792(12)	.812(9)
Ipums	8844	61	0.3	8736	.029(5)	.048(4)	.028(6)	.073(1)	.028(6)	.072(2)	.063(3)	.016(8)	.014(9)	.009(12)	.012(10)	.011(11)
Optdigits	5620	65	9.9	3622	.626(12)	.997(2)	.729(10)	.996(4)	.729(10)	.997(2)	.946(9)	.957(8)	.998(1)	.996(4)	.991(7)	.995(6)
Spambase	4601	58	39.4	206	.856(7)	.905(1)	.856(7)	.901(3)	.856(7)	.902(2)	.880(4)	.860(6)	.848(10)	.870(5)	.693(12)	.834(11)
Synthetic	600	62	16.7	266	.309(12)	1.0(1)	.356(10)	1.0(1)	.355(11)	1.0(1)	.936(5)	.935(6)	.650(9)	.940(4)	.679(8)	.876(7)
Waveform	5000	41	33.1	573	.686(9)	.862(1)	.667(10)	.862(1)	.667(10)	.862(1)	.658(12)	.700(8)	.812(6)	.841(4)	.796(7)	.816(5)
Agnostic-vs-Prior[8]:																
A.agnostic	4562	49	24.8	1157	.376(10)	.708(3)	.284(12)	.715(1)	.285(11)	.715(1)	.554(7)	.526(8)	.659(4)	.611(5)	.579(6)	.513(9)
A.prior	4562	15	24.8	1157	.512(8)	.643(1)	.457(10)	.638(2)	.457(10)	.638(2)	.608(4)	.487(9)	.552(6)	.530(7)	.565(5)	.416(12)
G.agnostic	3468	785	9.1	2322	.232(12)	.824(1)	.339(11)	.810(3)	.340(10)	.820(2)	.521(8)	.519(9)	.728(6)	.772(5)	.797(4)	.718(7)
H.prior	4229	1618	3.5	3654	.030(12)	.282(1)	.035(10)	.096(8)	.035(10)	.174(5)	.091(9)	.106(7)	.261(3)	.270(2)	.135(6)	.254(4)
S.agnostic	14395	217	6.2	11069	.690(12)	.923(1)	.750(11)	.759(9)	.752(10)	.782(8)	.821(7)	.846(6)	.914(3)	.916(2)	.912(4)	.907(5)
S.prior	14395	109	6.2	11069	.785(10)	.930(1)	.781(11)	.803(8)	.781(11)	.811(7)	.865(6)	.877(5)	.919(2)	.912(3)	.796(9)	.892(4)
Bio-Seville[9]:																
Colon	62	2001	35.5	5	.841(2)	.844(1)	.799(8)	.836(3)	.799(8)	.836(3)	.711(11)	.641(12)	.816(6)	.828(5)	.765(10)	.811(7)
Complete	96	4027	22.2	28	.813(2)	.802(3)	.797(4)	.796(7)	.797(4)	.797(4)	.531(11)	.531(11)	.819(1)	.778(8)	.769(9)	.753(10)
Tumor	60	7130	35.0	5	.428(9)	.445(5)	.453(1)	.453(1)	.453(1)	.453(1)	.353(11)	.337(12)	.433(7)	.437(6)	.403(10)	.429(8)
Text Mining [15]:																
fbis	2463	2001	1.5	2313	.029(10)	.103(6)	.029(10)	.144(4)	.028(12)	.190(3)	.216(2)	.241(1)	.069(9)	.099(7)	.132(5)	.093(8)
oh0	1003	3183	5.1	809	.060(10)	.398(5)	.059(11)	.311(8)	.059(11)	.498(2)	.498(2)	.452(4)	.509(1)	.385(6)	.213(9)	.371(7)
oh10	1050	3239	5.0	852	.075(10)	.217(2)	.074(11)	.129(7)	.074(11)	.217(2)	.107(9)	.127(8)	.289(1)	.178(5)	.211(4)	.137(6)
tr12	313	5805	9.3	207	.345(10)	.939(3)	.345(10)	.941(2)	.345(10)	.943(1)	.917(5)	.917(5)	.919(4)	.877(7)	.807(9)	.812(8)
tr23	204	5833	17.6	85	.437(12)	.667(5)	.439(10)	.686(3)	.439(10)	.685(4)	.763(1)	.763(1)	.663(6)	.652(7)	.556(9)	.635(8)
Average Rank					8.68	2.72	8.27	4.31	8.27	3.44	6.62	7.21	4.93	4.75	7.79	6.62
Friedman (Quadratic vs. Arithmetic)					✓6E-6		✓5E-5		✓2E-5		—	—	—	—	—	—
Friedman (All)					✓5.7E-6	**Base**	✓3E-5	0.117	✓3E-5	0.414	✓0.008	✓7E-4	✓0.003	✓0.003	✓3E-5	✓6E-4

loss functions is still convex. So Equation 4.7 is a *convex optimization* problem.

We use the bundle method [26] to solve Equation 4.7. Bundle method uses subgradients of the empirical risk function to approximate its piecewise linear lower bound (Figure 7 gives an example). By taking linearizations (i.e. first order Taylor approximation) on the empirical risk function, the lower bound is tightened iteratively until the difference gap between the approximated lower bound and the real risk function is smaller than a threshold ϵ. An advantage of the bundle method is that it only requires subgradients of the risk function ($\mathrm{R}^Q_{emp}(\mathbf{w})$) instead of the entire objective function ($J(\mathbf{w})$). Given ϵ as a convergence threshold (shown in Algorithm 1), this method is reported having an $O(\frac{1}{\epsilon})$ rate of convergency for non-smooth loss functions and $O(\log(\frac{1}{\epsilon}))$ for smooth functions [26].

Algorithm 1 lists the process of solving Equation 4.7. The subproblem of line 7 in Algorithm 1 is solved by a Fechnel dual formulation [18]. The subgradient of `QMLearn` empirical risk function used in Line 4 is computed as follows. For solving classification problems, define:

$$f(\mathbf{w}) = \frac{\sum_{i=1}^{\#pos} l(x_i, y_i, \mathbf{w})}{\#pos}$$

$$g(\mathbf{w}) = \frac{\sum_{i=\#pos+1}^{n} l(x_i, y_i, \mathbf{w})}{\#neg}$$

then the subgradient of $\mathrm{R}^Q_{emp}(\mathbf{w})$ is:

$$\begin{aligned} &\partial_{\mathbf{w}} R^Q_{emp}(\mathbf{w}) \\ =&\partial_{\mathbf{w}} \sqrt{\frac{f(\mathbf{w})^2 + g(\mathbf{w})^2}{2}} \\ =&\frac{1}{2}\left(\frac{f(\mathbf{w})^2 + g(\mathbf{w})^2}{2}\right)^{-\frac{1}{2}} (f(\mathbf{w})f'(\mathbf{w}) + g(\mathbf{w})g'(\mathbf{w})) \end{aligned} \tag{4.8}$$

The derivative of loss functions $l'(\mathrm{x}_i,\mathrm{y}_i,\mathbf{w})$ used in calculating $f'(\mathbf{w})$ and $g'(\mathbf{w})$ are listed in the last column of Table 1. The subgradients of `QMLearn`-based risk functions for regression estimation (Equation 3.4) are derived in the same way which we omit here.

Table 3: Comparisons between sampling methods and QMLearn models. The classifiers whose names end with "-S" are trained on data balanced by Safe-level-SMOTE method, while the QMLearn methods are trained on original imbalanced data. "S vs. Q" in the first row of Friedman tests denotes the comparisons between the over-sampling methods and QMLearn methods.

Datasets	Area Under Precision-Recall Curve					
	LogR-S	LogR^Q	Hinge-S	Hinge^Q	Huber-S	Huber^Q
Appetency	.032(2)	.058(1)	.031(3)	.026(6)	.031(3)	.027(5)
Churn	.093(3)	.123(1)	.088(5)	.093(3)	.088(5)	.101(2)
Upselling	.275(2)	.309(1)	.272(3)	.142(6)	.272(3)	.146(5)
Anneal	.239(6)	.478(3)	.266(4)	.765(1)	.266(4)	.763(2)
Arrhythmia	.068(1)	.052(4)	.065(2)	.043(5)	.065(2)	.043(5)
Cylinder	.610(4)	.735(1)	.593(6)	.718(2)	.594(5)	.718(2)
German	.496(4)	.613(3)	.489(5)	.624(1)	.489(5)	.623(2)
Hypothyroid	.033(4)	.737(3)	.033(4)	.859(2)	.033(4)	.864(1)
Ipums	.017(1)	.007(4)	.016(2)	.005(5)	.016(2)	.005(5)
Ionosphere	.809(4)	.849(1)	.809(4)	.821(2)	.809(4)	.821(2)
Optdigits	.996(3)	.997(1)	.996(3)	.996(3)	.996(3)	.997(1)
Spambase	.788(4)	.905(1)	.787(5)	.901(3)	.787(5)	.902(2)
Synthetic	.297(6)	1.0(1)	.298(4)	1.0(1)	.298(4)	1.0(1)
Waveform	.761(6)	.862(1)	.763(4)	.862(1)	.763(4)	.862(1)
A.agnostic	.606(6)	.708(3)	.635(4)	.715(1)	.635(4)	.715(1)
A.prior	.459(4)	.643(1)	.454(5)	.638(2)	.453(6)	.638(2)
G.agnostic	.631(4)	.824(1)	.626(5)	.810(3)	.626(5)	.820(2)
H.prior	.237(6)	.282(3)	.265(4)	.296(1)	.265(4)	.287(2)
S.agnostic	.902(2)	.923(1)	.880(3)	.759(6)	.880(3)	.782(5)
S.prior	.904(2)	.930(1)	.883(3)	.803(6)	.883(3)	.811(5)
Colon	.786(6)	.844(3)	.855(1)	.836(4)	.855(1)	.836(4)
Complete	.834(1)	.802(2)	.797(3)	.796(6)	.797(3)	.797(3)
Tumor	.418(6)	.445(5)	.453(1)	.453(1)	.453(1)	.453(1)
fbis	.032(4)	.103(3)	.031(5)	.144(2)	.031(5)	.190(1)
oh0	.618(3)	.398(5)	.622(1)	.311(6)	.622(1)	.498(4)
oh10	.359(1)	.217(4)	.358(2)	.129(6)	.358(2)	.217(4)
tr12	.897(4)	.939(3)	.893(5)	.941(2)	.893(5)	.943(1)
tr23	.657(4)	.667(3)	.635(5)	.686(1)	.634(6)	.685(2)
Average Rank	3.58	2.24	3.51	3.06	3.55	2.55
Friedman (S vs. Q)	✓6E-4		0.432		0.239	
Friedman (All)	✓6E-4	**Base**	✓0.003	✓0.049	✓0.002	0.221

5 Experiments and Analysis

In this section, we analyze and compare the performance between existing statistical learners and QMLearn-based learners. We conduct experiments separately on problems of classifications and regressions. Besides comparing with existing methods, in the experiments of classification we also compare QMLearn-based methods against other *algorithm-oriented* approaches (i.e. CCPDT[2] and HDDT[3]) and *data-oriented* approaches (i.e. SDC, GP, safe-level-SMOTE). To make the effectiveness of our method more convincing, we also include the recently proposed SVM models Liblinear [12] and ASVM [32] in evaluations.

The safe-level-SMOTE method is implemented on the basis of the Weka [31] version of SMOTE algorithm. Decision trees CCPDT and HDDT are pruned by Fisher's exact test (as recommended in [23]). All experiments are carried out using 5×2 folds cross-validations, and the final results are the average of the repeated runs. We set λ in Equation 2.1 and 4.7 to $\frac{1}{2}$ constantly[4]. The bundle method convergence threshold ϵ (as suggested in [26]) is set to 1E-5.

5.1 Classification In this subsection, we report the experiments on logistic regression [9], SVM with hinge loss [13] and SVM with Huber loss [4]. We use the same parameters of loss functions between conventional learners and QMLearn learners. The Huber loss threshold (i.e. parameter h in the Huber-SVM loss function shown in Table 1) is set to 0.01. We select 28 data sets from KDDCup'09[5], UCI repository [2], agnostic vs. prior competition[6], bio-seville[7], and text mining domain [15]. For multiple-label data sets, we keep the smallest label as the positive class, and combine all the other labels as the negative class. Details of the data sets are shown in Table 2. Beside the proportion of the minor class in a data set, we also present the coefficient of variation (CV) [17] to measure imbalance. CV is defined as the ratio of the standard deviation and the mean of the class counts in data sets. So the greater the value of CV, the more imbalanced the data set.

The metric of AUC-PR (area under precision-recall curve) has been reported in [10] better than AUC-ROC (area under ROC curve) for handling imbalanced data. A curve dominates in ROC space if and only if it dominates in PR space, and classifiers that are more superior in terms of AUC-PR are definitely more superior in terms of AUC-ROC, but not vice versa [10]. Hence we use the more informative metric of AUC-PR for classifier comparisons. The performance of existing statistical learners and QMLearn-based learners are shown in Table 2.

While there are various ways to compare classifiers across multiple data sets, we adopt the strategy proposed by [11] which evaluates classifiers by ranks. In Table 2 the classifiers in comparison are ranked on each data set by the value of their AUC-PR, with ranking of 1 being the best. We carry out Friedman tests on the sequences of ranks between different classifiers. In the Friedman tests, p–values that are lower than 0.05 reject the hypothesis with 95% confidence that the classifiers in comparison are not statistically different.

As shown in the bottom of Table 2, we first take the Friedman tests within the same type of classifiers, i.e. "LogR vs. LogR^Q", "Hinge vs. Hinge^Q", and

[2]Code from www.cs.usyd.edu.au/~weiliu/CCPDT_src.zip

[3]Code from www.nd.edu/~dial/software/hddt.tar.gz

[4]Here we do not investigate the effect of λ on the convergence of bundle methods, since this evaluation has been reported in the methods' original paper [26].

[5]http://www.kddcup-orange.com/data.php

[6]http://www.agnostic.inf.ethz.ch

[7]http://www.upo.es/eps/bigs/datasets.html

Table 4: Data sets with categorical classes and regression comparisons among different regression methods. "Skewness" is measured by the Pearson's second skewness coefficient.

Datasets	#Inst	#Attr	Skewness	Root Mean Square Error					
				LR	LR^Q	SVR	SVR^Q	QR	QR^Q
StatLib[10]:									
Analcat	4052	8	-1.4539	.361(5)	.259(2)	.31(4)	.241(1)	.423(6)	.260(3)
Bodyfat	252	15	-0.2257	.065(5)	.037(3)	.073(6)	.060(4)	.029(2)	.026(1)
Boston	506	14	0.4347	.199(6)	.139(3)	.180(4)	.128(2)	.190(5)	.113(1)
Diggle	310	9	-0.3275	.120(5)	.086(3)	.113(4)	.083(2)	.121(6)	.043(1)
Houses	20640	9	-1.6167	.291(4)	.411(6)	.173(1)	.274(3)	.333(5)	.201(2)
Space	3107	7	-0.0348	.191(6)	.122(3)	.14(4)	.112(2)	.158(5)	.110(1)
Wind	6574	15	0.2685	.125(6)	.096(3)	.107(4)	.092(2)	.111(5)	.087(1)
UCI [2]:									
Ailerons	13750	41	-0.5287	.095(6)	.078(3)	.078(3)	.076(2)	.080(5)	.058(1)
Bank	8192	33	1.4046	.149(5)	.138(3)	.148(4)	.134(2)	.163(6)	.129(1)
Cal	20640	9	0.7068	.237(5)	.211(3)	.236(4)	.195(1)	.252(6)	.196(2)
Cleveland	303	14	2.2888	.252(4)	.226(2)	.253(5)	.226(2)	.280(6)	.220(1)
Planes	40768	11	-0.0082	.217(4)	.221(6)	.201(2)	.211(3)	.219(5)	.189(1)
Elevators	16599	19	0.7258	.160(6)	.116(3)	.123(4)	.101(2)	.123(4)	.089(1)
Fried	40768	11	-0.0097	.148(3)	.295(6)	.121(1)	.244(5)	.133(2)	.199(4)
Meta	528	22	0.3518	.062(1)	.062(1)	.082(5)	.086(6)	.064(4)	.063(3)
Mv	40768	11	-1.1044	.194(3)	.223(6)	.180(1)	.211(4)	.216(5)	.180(1)
Pharynx	195	12	0.8077	.211(5)	.161(2)	.201(4)	.16(1)	.214(6)	.161(2)
Pol	15000	49	2.0811	.417(4)	.387(2)	.458(5)	.374(1)	.507(6)	.392(3)
Puma	8192	33	0.0171	.312(4)	.299(1)	.312(4)	.301(2)	.32(6)	.302(3)
Quake	2178	4	1.2223	.782(5)	.675(1)	.78(4)	.733(2)	.815(6)	.770(3)
Stock	950	10	0.1122	.131(6)	.091(3)	.104(4)	.083(2)	.125(5)	.066(1)
Average Rank				4.5	3.0	3.54	2.36	4.86	1.72
Friedman (Quadratic vs. Arithmetic)				✓.0073		✓.0164		✓3.4E-5	
Friedman (All)				✓2.1E-4	✓.0253	✓3.5E-4	✓.0495	✓3.4E-5	**Base**

"Huber vs. Huber^Q". We use LogR, Hinge, Huber to denote logistic regression and SVM with hinge loss and Huber loss functions, while LogR^Q, Hinge^Q and Huber^Q represent those traditional classifiers applied with `QMLearn`. From the low p–values, it is easy to see that `QMLearn`-based learners are always better than their corresponding traditional learners. After that we also compare the best classifier we obtain (i.e. LogR^Q) with CCPDT, HDDT, SDC, GP, Liblinear and ASVM, where LogR^Q significantly outperforms all of these classifiers.

5.2 Effects of Safe-level-SMOTE Sampling In this experiment we compare `QMLearn` model trained on original imbalanced data with existing learners trained on data balanced by the safe-level-SMOTE technique [3] (which we denote by "method name -S"). The comparison results are reported in Table 3.

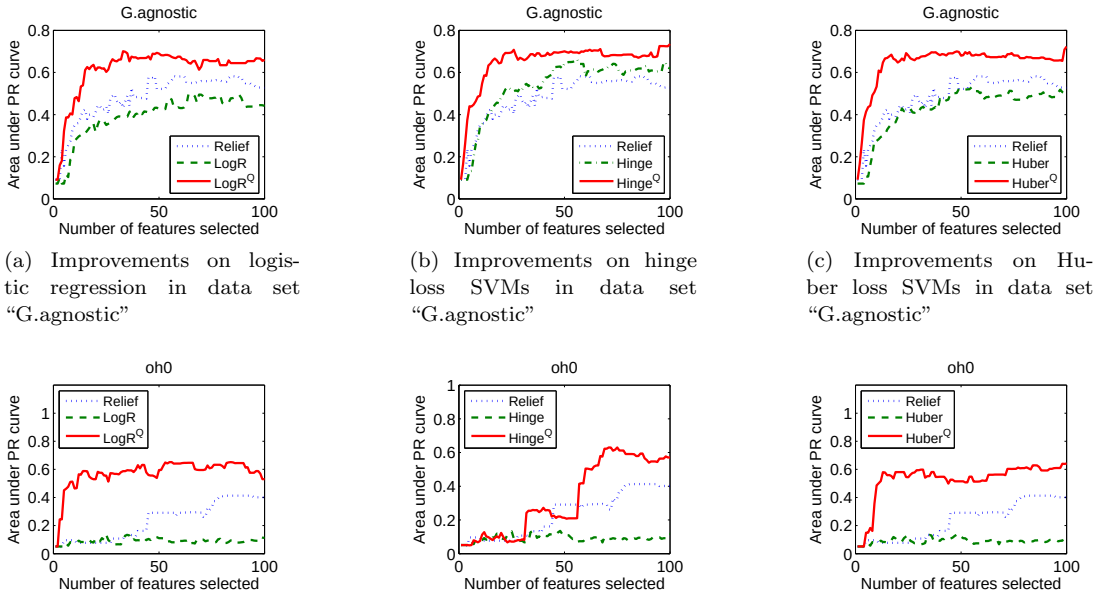
We obtain the best results from LogR^Q which is significantly better than LogR-S, while Hinge^Q and Huber^Q is comparable to (better but not significant than) Hinge-S and Huber-S. This observation suggests that if one uses `QMLearn` method he can obtain results comparable to the cutting-edge sampling technique, so the extra computational cost of data sampling before training can potentially be saved.

5.3 Regression In this subsection we present the results of regression estimation obtained from ordinary linear regression [30] (LR), support vector regression (SVR) [28] and quantile regression (QR) [20]. In SVR we set the insensitive threshold (ϵ in the SVR loss function of Table 1) to 0.01. In QR we use 50% percentile (i.e. τ in the quantile regression loss function of Table 1 is set to 0.5).

We select 21 numerical-class data sets from StatLib[8] and UCI [2] shown in Table 4. The last column of the table is the skewness of the numerical class variable which is measure by Pearson's second skewness coefficients. This skewness measure is defined as $\frac{3(\text{mean}-\text{median})}{\text{standard deviation}}$. We use root mean squared error (RMSE) as the metric of comparisons in this experiment. The performance of the three pairs of learners are listed in Table 4. The low p-values from the Friedman tests of all three pairs demonstrate that `QMLearn` models statistically outperforms the existing regression models. Notably, `QMLearn`-based quantile regression model significantly outperforms all the other five models. Since quantile regression is originally resilient to noises, the `QMLearn`-based 50% quantile regression becomes the most robust regression model in handling skewed data.

5.4 Feature Selection The weight vectors obtained from traditional empirical risk functions and `QMLearn`-based empirical risk functions give different feature rankings, which results in different feature selection performance. In this subsection we compare the usefulness of features selected by these two types of methods.

[8] `http://lib.stat.cmu.edu/`



(a) Improvements on logistic regression in data set "G.agnostic"

(b) Improvements on hinge loss SVMs in data set "G.agnostic"

(c) Improvements on Huber loss SVMs in data set "G.agnostic"

(d) Improvements on logistic regression in data set "oh0"

(e) Improvements on hinge loss SVMs in data set "oh0"

(f) Improvements on Huber loss SVMs in data set "oh0"

Figure 8: Feature selection comparisons on *categorical-class* data sets "G.agnostic" and "oh0". Features with highest weights (top ranked features) from `QMLearn` methods (shown by solid red lines) are always more effective than those from existing methods and Relief. Subfig (d) and (f) are prominent examples illustrating the difference of existing method and `QMLearn` method – while the features ranked the highest by `QMLearn` methods generate high AUC-PR values, the features that existing methods select are mostly trivial ones.

In order to decouple the effects of learning algorithms where feature weights are generated, we use other algorithms that are not included in the training process to evaluate features: when examining the effectiveness of features from categorical-class data sets, we use the CCPDT decision tree algorithm, and for those from numerical-class data sets we use Gaussian process regression [25].

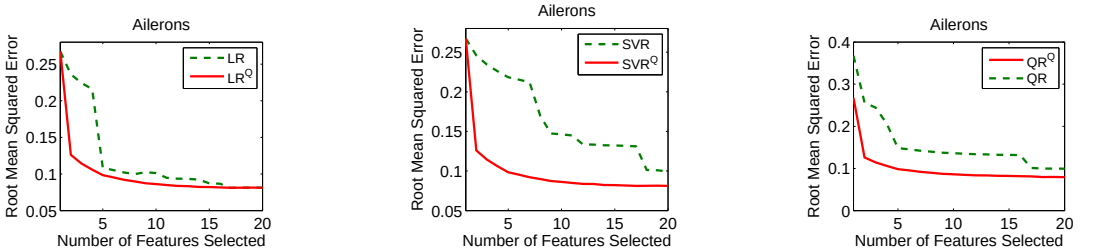
Since the evaluation metrics (AUC-PR/RMSE) will be very similar among different feature rankings if the number of features selected is close to the total number of features, we only look at the performance of the features ranked relatively higher. So for data sets with more than 200 features, we examine their top 100 ranked features; and for other data sets we test the top half of their entire feature sets.

When we analyze the performance of `QMLearn` methods on categorical-class data sets, we also compare them with another commonly used feature selection technique – Relief (relevance in estimating features) [19]. Due to page limits, here we only present the feature selection comparisons on two categorical-class data sets and two numerical-class data sets shown in Figure 8 and 9.

In the case of categorical-class data sets (Figure 8), `QMLearn` features are significantly better than both Relief and existing methods. Here it is not necessary to carry out Friedman tests of rankings to prove the significance, since there are seldom overlaps between the `QMLearn` method (solids lines) and the existing methods (dashed lines). In the case of numerical-class data sets (Figure 9) we observe some "ties" besides the "wins" of `QMLearn` over existing regression models. This is because the total number of features in these data sets are relatively small and hence there are smaller feature ranking differences compared to the ones from categorical-class data. But it does not effect the illustration of the advantage of `QMLearn` over existing methods in feature selection.

6 Conclusions and future research

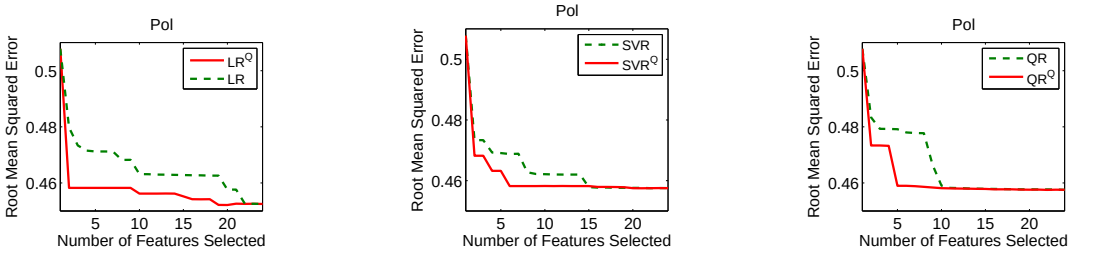
The main focus of this paper is to build statistical machine learners that are robust and insensitive to class skewness/imbalance. The traditional empirical risk function uses a flawed arithmetic mean of scalar losses. We have shown that such use of arithmetic mean makes machine learners insensitive to prediction errors of the minor class.



(a) Improvements on ordinal linear regression in data set "Ailerons"

(b) Improvements on SVR in data set "Ailerons"

(c) Improvements on quantile regression in data set "Ailerons"



(d) Improvements on ordinal linear regression in data set "Pol"

(e) Improvements on SVR in data set "Pol"

(f) Improvements on quantile regression in data set "Pol"

Figure 9: Feature selection comparisons on *numerical-class* data sets "Ailerons" and "Pol". The feature ranks from LR^Q, SVR^Q and QR^Q are similar in the top 20 features of data set "Ailerons", and hence their performance are similar (shown by the red solid lines in subfig. (a), (b) and (c)). This observation suggests that the three `QMLearn` methods are all able to identify the most effective features from their learned weight vectors, which the existing methods fail to achieve.

We have proposed `QMLearn`, a novel framework which redefines the empirical risk function by using the notion of quadratic mean. `QMLearn` has been shown theoretically and empirically insensitive to class imbalance/skewness. We have applied this framework into logistic regression, SVM, linear regression, SVR and quantile regression, all of which prove the superiority of the `QMLearn` method.

On classification problems `QMLearn` is not only significantly better than existing methods, but also comparable to the state-of-the-art sampling technique. This observation suggests the extra computational cost of data sampling before training can be saved if `QMLearn` is employed.

Another advantage of `QMLearn` is that, the weight vectors obtained form `QMLearn` models are significantly better than those from existing methods for feature selection when data is skewed/imbalanced. It is important to identified the features that are the most resilient to class skewness/imbalance, and this information is revealed from the `QMLearn`-based weight vectors.

More importantly, we consider `QMLearn` promising and theoretically valuable since it is a generic framework, and by changing the loss functions it can be applied to lots of other existing statistical learners such as Gaussian Processes and Conditional Random Fields (CRFs) etc.

In future our plan is to explore the use of `QMLearn` methods on non-linear machine learners, and investigate the effects of kernel tricks when facing skewed/imbalanced data distributions.

Acknowledgements

The first author of this paper acknowledges the financial support of the Capital Markets CRC.

References

[1] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. *Lecture Notes in Computer Science: ECML 2004*, 3201:39–50.

[2] A. Asuncion and D.J. Newman. UCI Machine Learning Repository, 2007.

[3] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap. Safe-Level-SMOTE: Safe-Level-

Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem . *Advances in Knowledge Discovery and Data Mining (PAKDD'09)*, 5476:475–482, 2009.

[4] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, 2007.

[5] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer. SMOTE: synthetic minority oversampling. *Journal of Artificial Intelligence Research*, 16(1):321–357, 2002.

[6] N.V. Chawla, N. Japkowicz, and A. Kotcz. Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004.

[7] N.V. Chawla, A. Lazarevic, L.O. Hall, and K.W. Bowyer. SMOTEBoost: Improving prediction of the minority class in boosting. *Lecture notes in computer science*, pages 107–119, 2003.

[8] D.A. Cieslak and N.V. Chawla. Learning Decision Trees for Unbalanced Data. In *Lecture Notes of Computer Science: ECML PKDD 2008 Part I*, pages 241–256, 2008.

[9] M. Collins, R.E. Schapire, and Y. Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 48(1):253–285, 2002.

[10] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 233–240, 2006.

[11] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[12] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[13] C. Gentile and M.K. Warmuth. Linear hinge loss and average margin. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 225–231.

[14] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002.

[15] E.H. Han and G. Karypis. Centroid-based document classification: Analysis and experimental results. *Principles of Data Mining and Knowledge Discovery*, pages 116–123, 2000.

[16] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer, 2001.

[17] W.A. Hendricks and K.W. Robey. The sampling distribution of the coefficient of variation. *The Annals of Mathematical Statistics*, 7(3):129–132, 1936.

[18] J.B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms: Fundamentals*. Springer, 1993.

[19] K. Kira and L.A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the National Conference on Artificial Intelligence*, pages 129–129, 1992.

[20] R. Koenker. *Quantile regression*. Cambridge University Press, 2005.

[21] S. Köknar-Tezel and L.J. Latecki. Improving svm classification on imbalanced data sets in distance spaces. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, pages 259–267, 2009.

[22] B. Liu, W. Hsu, Y. Ma, A.A. Freitas, and J. Li. Integrating Classification and Association Rule Mining. *IEEE Transactions on Knowledge and Data Engineering*, 18:460–471.

[23] W. Liu, S. Chawla, D.A. Cieslak, and N.V. Chawla. A Robust Decision Tree Algorithms for Imbalanced Data Sets. In *Proceedings of the Tenth SIAM International Conference on Data Mining*, pages 766–777, 2010.

[24] K.M. McGreevy, S.R. Lipsitz, J.A. Linder, E. Rimm, and D.G. Hoel. Using median regression to obtain adjusted estimates of central tendency for skewed laboratory and epidemiologic data. *Clinical chemistry*, 55(1):165–169, 2009.

[25] M. Seeger, C.K.I. Williams, and N.D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *Workshop on AI and Statistics*, volume 9.

[26] C.H. Teo, SVN Vishwanthan, A.J. Smola, and Q.V. Le. Bundle methods for regularized risk minimization. *Journal of Machine Learning Research*, 11:311–365, 2010.

[27] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[28] V. Vapnik, S.E. Golowich, and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. *Advances in neural information processing systems*, pages 281–287, 1997.

[29] F. Verhein and S. Chawla. Using Significant, Positively Associated and Relatively Class Correlated Rules for Associative Classification of Imbalanced Datasets. In *Seventh IEEE International Conference on Data Mining*, pages 679–684, 2007.

[30] C.K.I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. *Learning in graphical models*, pages 599–621, 1997.

[31] I.H. Witten and E. Frank. Data mining: practical machine learning tools and techniques with Java implementations. *ACM SIGMOD Record*, 31(1):76–77, 2002.

[32] S.H. Wu, K.P. Lin, C.M. Chen, and M.S. Chen. Asymmetric support vector machines: low false-positive learning. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 749–757, 2008.

[33] Q. Yang and X.D. Wu. 10 challenging problems in data mining research. *International Journal of Information Technology and Decision Making*, 5(4):597–604, 2006.

[34] K. Yu, Z. Lu, and J. Stander. Quantile regression: applications and current research areas. *Journal of the Royal Statistical Society*, 52(3):331–350.