2010 IEEE International Conference on Data Mining Workshops

From Convex to Nonconvex: a Loss Function Analysis for Binary Classification

Lei Zhao
GSITMS, University of Ballarat
Ballarat, Australia
Email: l.zhao@ballarat.edu.au

Musa Mammadov
GSITMS, University of Ballarat
Ballarat, Australia
Email: m.mammadov@ballarat.edu.au

John Yearwood
GSITMS, University of Ballarat
Ballarat, Australia
Email: j.yearwood@ballarat.edu.au

***Abstract*—Problems of data classification can be studied in the framework of regularization theory as ill-posed problems. In this framework, loss functions play an important role in the application of regularization theory to classification. In this paper, we review some important convex loss functions, including hinge loss, square loss, modified square loss, exponential loss, logistic regression loss, as well as some non-convex loss functions, such as sigmoid loss, ϕ-loss, ramp loss, normalized sigmoid loss, and the loss function of 2 layer neural network. Based on the analysis of these loss functions, we propose a new differentiable non-convex loss function, called smoothed 0-1 loss function, which is a natural approximation of the 0-1 loss function. To compare the performance of different loss functions, we propose two binary classification algorithms for binary classification, one for convex loss functions, the other for non-convex loss functions. A set of experiments are launched on several binary data sets from the UCI repository. The results show that the proposed smoothed 0-1 loss function is robust, especially for those noisy data sets with many outliers.**

***Index Terms*—classification; optimization; non-convex; loss function; regularization**

I. Introduction

The purpose of *Supervised Learning* is to *learn* or *train* a function f_S by given training set S. In this paper we consider binary classification where the l training examples satisfy $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}$ for all i. By using f_S we can predict the corresponding label y_{new} of a new point $\mathbf{x}_{new}$, where $y_{new} = 1$ for $f_S(x_{new}) > 0$ and $y_{new} = -1$ otherwise. Most of the discussions in this paper can be directly applied to non-linear kernel classifiers [1]. Without losing generality, in this paper we focus on learning linear functions $f_S(\mathbf{x}) = \mathbf{x} \cdot \mathbf{w} + b$ for binary classification problems from training set S. For the sake of simplicity, we denote f_S by f.

The function f actually is a linear combination of dictionary functions coming from a dictionary $\mathcal{H}$ which can be large or even infinite. When $|\mathcal{H}|$ is large, some regularization is needed to control the "complexity" of the function f and the resulting overfitting. Actually, supervised learning is commonly studied in the framework of Regularization Theory as ill-posed problems, or through Statistical Learning Theory in the learning from example paradigm. The connection between these two approaches has been discussed by Evgeniou et. al. [2], [3].

Inspired by Evgeniou et. al. [2], [3], this paper focuses on the following Tikhonov regularization framework:

$$\min_{f \in \mathcal{H}} H[f] = \frac{1}{l} \sum_{i=1}^{l} V(f(\mathbf{x}_i), y_i) + \lambda ||f||_K^2 \qquad (1)$$

where V denotes the *loss function*, $\|f\|_K^2$ is the norm of f squared measured in a Reproducing Kernel Hilbert Space (see [4]), $\frac{1}{l} \sum_{i=1}^{l} V(f(\mathbf{x}_i), y_i)$ measures the *empirical error* of the corresponding f, and λ is called *regularization constant* that control the tradeoff between empirical error and regularization effects. In this problem, $\|f\|_K^2$ is differentiable and cheap to compute. Contrarily, the empirical error can be non-differentiable, non-convex, and computationally expensive to deal with.

We can achieve different learning schemes simply by varying loss function V or the norm of f. This regularization framework is based on the assumption that there exists an unknown function $f : X \to Y$ that provides a labeling $y \in Y$ for a given $\mathbf{x} \in X$. Tikhonov regularization attempts to find this unknown function which simultaneously has small empirical error on a training set and small norm in a Reproducing Kernel Hilbert Space.

The central question of classification is how well the chosen function generalizes, or how well it estimates the output for previously unseen inputs [5]. To evaluate f, we define loss function $V(f(\mathbf{x}_i), y_i)$ of a given example $(\mathbf{x}_i, y_i)$, which measures the "goodness" of the predicted output $f(\mathbf{x}_i)$ with respect to the given output y_i.

For binary classification, the most straightforward loss function is the 0-1 loss function:

$$V(f(\mathbf{x}), y) = \ominus(-yf(\mathbf{x})) \qquad (2)$$

where $\ominus(z) = 0$ for $z < 0$ and $\ominus(z) = 1$ otherwise. This is an "ideal" loss function, making as few mistakes as possible. However, trying to optimize the 0-1 loss directly leads to non-convex optimization problem. Moreover it is not continuous, insensitive to the magnitude of f, and regularization of f is meaningless. Therefore, a number of surrogate loss functions are proposed in the literature.

Broadly speaking, loss functions can be divided into two categories: convex loss functions and non-convex loss functions. Convex loss functions including hinge loss, square loss

978-0-7695-4257-7/10 $26.00 © 2010 IEEE
DOI 10.1109/ICDMW.2010.57

are the most commonly used. The convexity of these loss functions are viewed as highly preferable in many publications because of their computational advantages (unique optima, ease-of-use, ability to be efficiently optimized by convex optimization tools, etc.). However, the convexity also offer poor approximations to the 0-1 loss function and lack of robustness to outliers due to their boundlessness, which makes the corresponding classifier liable to be dominated by outliers. Therefore, different non-convex loss functions, such as ramp loss function and sigmoid loss function, are proposed recently.

Hastie et al. [6] compare different convex loss functions for SVM, LLSF, LR and AdaBoost, in a way such that the sensitivity of those methods with respect to outliers. Fan Li and Yiming Yang [7] make a loss function based study with respect to eight classifiers popular in text categorization, including SVM, linear regression, logistic regression, neural networks, Rocchio-style, Prototypes, kNN and Nave Bayes. However, the differences between convex and non-convex loss functions are not discussed, and some popular non-convex loss functions, such as ramp loss, sigmoid loss are not considered.

It would be valuable to launch a formal analysis of a broader range of loss functions on the optimization criterion (see Section II). The main contribution of this paper is that, in the framework of Tikhonov regularization, we propose a new non-convex loss function, called smoothed 0-1 loss function. To compare the performance of different loss functions, two binary classification algorithms are proposed. A set of experiments are launched on several binary data sets from the UCI repository. The results show that the proposed smoothed 0-1 loss function is robust, especially for those noisy data sets with many outliers.

The organization of the remaining parts of this paper is as follows: Section II reviews some important convex and non-convex loss functions. In Section III we propose a new non-convex loss function: smoothed 0-1 loss function. In Section IV, an optimization algorithm (QSM) that suitable for both smooth and non-smooth optimization problems are adopted, which makes it possible to compare different loss functions under the same framework. In Section V we develop two binary classification algorithms for both convex and non-convex loss functions. Section VI describes the experimental settings and results. We summarize and make conclusion in Section VII.

II. A Review of Loss Functions

In this section, we review different loss functions from the viewpoint of convexity.

A. Convex Loss Functions

In the literature, loss functions are commonly assumed to be convex [8]. The main advantage of this type of loss functions is the computational simplicity, and complex global optimization approaches can be avoided. Square loss and hinge loss are the most commonly adopted loss functions in machine learning.

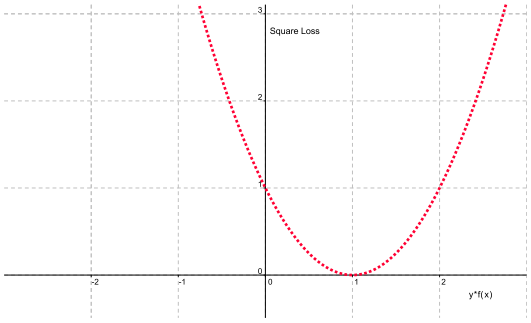

Fig. 1. The square loss $V(f(\mathbf{x}), y) = (1 - y \cdot f(\mathbf{x}))^2$

1) Square Loss: Among the convex loss functions, square loss function $V(f(\mathbf{x}), y) = (y - f(\mathbf{x}))^2$ is the most inexpensive one since solutions can be obtained merely through solving linear equations. This outstanding feature in computational efficiency makes square loss an appealing tool for mining large data sets [7].

By adopting square loss function, we can have a Regularized Least Squares Classification (RLSC) [9] model. It is also called Regularization networks in ([2]), and Proximal Support Vector Machine in ([10]). To put it in the same framework as SVMs, a simple transformation [7] is made for the square loss for binary classification ($y \in \{-1, 1\}$):

$$\begin{aligned} V(f(\mathbf{x}), y) &= (y - f(\mathbf{x}))^2 \\ &= y^2 - 2yf(\mathbf{x}) + (f(\mathbf{x}))^2 \\ &= 1 - 2yf(\mathbf{x}) + (yf(\mathbf{x}))^2 \\ &= (1 - yf(\mathbf{x}))^2 \end{aligned} \tag{3}$$

The corresponding linear system is defined as $f(\mathbf{x}) = \mathbf{x} \cdot \mathbf{w}$. Accordingly, we have the square loss function (see Figure 1), and the regularized least squares classification model (4).

$$\min_{\mathbf{w},b} \frac{1}{l} \sum_{i=1}^{l} (1 - y_i(\mathbf{x}_i \cdot \mathbf{w}))^2 + \lambda ||\mathbf{w}||^2 \tag{4}$$

From Figure 1, we can see that high penalty are given to those misclassified examples far from the origin, which make the corresponding model liable to be dominated by the outliers. Even worse, different from other loss functions, square loss function is not monotonically decreasing, which also heavily penalize those correctly classified examples with large positive value of $yf(\mathbf{x})$.

2) Hinge Loss: The classical SVM arises by considering *hinge loss* (see Figure 2).

$$V(f(\mathbf{x}), y) = (1 - y \cdot f(\mathbf{x}))_+ \tag{5}$$

where $(k)_+ \equiv \max(k, 0)$. The hinge loss has a simple but compelling justification [9]. If $y_i f(\mathbf{x}_i) \geq 1$, we pay no penalty for example i. If $y_i f(\mathbf{x}_i) < 1$, a penalty linear in the amount we fail to satisfy the constraint will be mounted. An important feature of the hinge loss is that it is an upper bound on the 0-1 loss, and thus the large-margin generalization error bounds

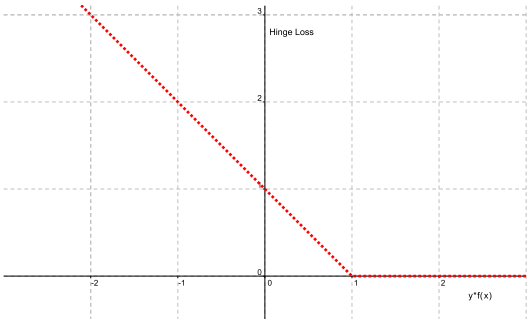

Fig. 2. The hinge loss $V(f(\mathbf{x}), y) = (1 - y \cdot f(\mathbf{x}))_+$

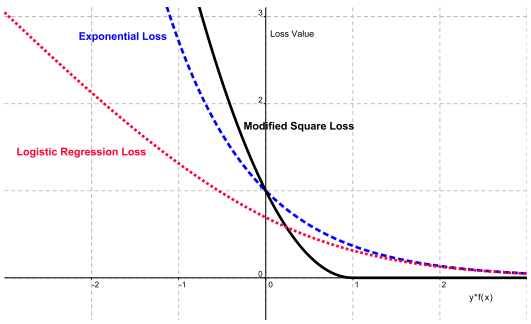

Fig. 3. Modified square loss function, Exponential loss function, and Logistic regression loss function

bounding its value on examples not in the training set also bounds the value of the 0-1 misclassification error [11].

Using the hinge loss, we have the following regularization problem:

$$\min_{f \in \mathcal{H}} \frac{1}{l} \sum_{i=1}^{l} (1 - y_i f(\mathbf{x}_i))_+ + \lambda ||f||_K^2 \tag{6}$$

If we consider f is only from linear function space and use slack variables ξ_i, corresponding to the penalty we pay at data point i, the problem becomes:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \tfrac{1}{l} \textstyle\sum_{i=1}^{l} \xi_i + \lambda \|\mathbf{w}\|^2 \\ s.t. \quad & y_i f(\mathbf{x}_i) \geq 1 - \xi_i \qquad i = 1, \cdots, l \\ & \xi_i \geq 0 \qquad\qquad\quad i = 1, \cdots, l \end{aligned} \tag{7}$$

where $\mathbf{w}$ and b is from the linear function $f(\mathbf{x}) = \mathbf{x} \cdot \mathbf{w} + b$,

In SVMs, the support vectors represent the most informative data points and compress the information contained in the training set: for classification, only the support vectors need to be stored, while all other training examples can be discarded. This, along with some geometric properties of SVMs such as the interpretation of the RKHS norm of their solution as the inverse of the margin (Vapnik, 1998), is a key property of SVM and might explain why this technique works well in many practical applications [5]. However, the foundation of SVM's margin theory becomes much less solid in non-separable cases [12].

3) Smoothed Hinge Loss: A difficulty with the hinge loss is that direct optimization is difficult, due to the discontinuity in the derivative at $z = 1$. Rennie and Srebro [11] proposed a smooth version of the Hinge (smoothed hinge loss).

$$V(f(\mathbf{x}), y) = \begin{cases} 0 & \text{if } yf(\mathbf{x}) \geq 1 \\ (1 - yf(\mathbf{x}))^2/2 & \text{if } 0 < yf(\mathbf{x}) \leq 1 \\ 0.5 - yf(\mathbf{x}) & \text{if } yf(\mathbf{x}) \leq 0 \end{cases} \tag{8}$$

Smoothed hinge loss preserves important features of the hinge loss, and is easier to minimize by using direct derivative.

4) Modified Square Loss: Similar to hinge loss, Zhang and Oles [13] propose a modified square loss (9) with smooth derivative.

$$V(f(\mathbf{x}), y) = \max(1 - yf(\mathbf{x}), 0)^2 \tag{9}$$

Compared with the hinge loss and the modified hinge loss, the modified square loss is much more sensitive to outliers and large errors.

5) Some Other Convex Loss Functions: There are some other popular loss functions, for example, the exponential loss function (10) used by Adaboost [14], and the log loss function (11) employed by Logistic Regression [15] (see Figure 3).

$$V(f(\mathbf{x}), y) = \exp(-yf(\mathbf{x})) \tag{10}$$

$$V(f(\mathbf{x}), y) = \ln(1 + \exp(-yf(\mathbf{x}))) \tag{11}$$

All of the above loss functions have been used in practical applications [16]. The common character of these loss functions is that all of them are convex, which make their corresponding regularization classification models easy to compute.

Unfortunately, in real application, data sets tend to be non-linearly separable. The drawback of convex loss function is that outliers are guaranteed to play a maximal role in determining the decision hyperplane, or in other words, the decision hyperplane is dominated by outliers. The reason is outliers tend to have very large margin loss.

In the literature, there have been a few attempts to improve the robustness of training to outliers. One attempt is a direct approach by formulating outlier detection and removal directly from the data sets . Most of these works focus on unsupervised learning [17], [18], while [19] focuses on supervised case. The other attempt sets a upper bound and make the loss stop increase after a certain point, which is also called non-convex loss function [20], [21], [22], [23].

B. Non-convex Loss Functions

In the literature, machine learning applications seem to have trouble moving beyond convex loss function models, i.e. regularized least squares classification, logistic regression, SVMs, and exponential-family graphical models. For a new machine learning model, convexity is viewed as a virtue. None mention the potential computational advantages of non-convex optimization, simply because everyone assumes that convex optimization is easier. Most authors warn the reader about the potentially high cost of non-convex optimization.

However, in some sense, loss should be intuitively bounded. We should not pay an infinite cost for misclassifying any one

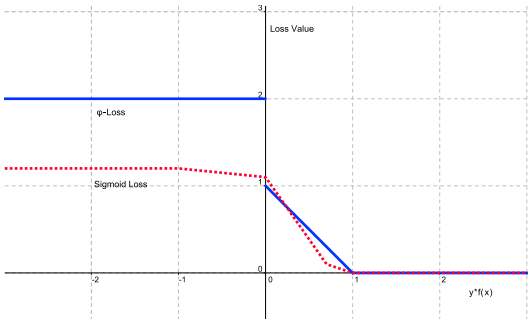

Fig. 4. ϕ-Loss and Sigmoid Loss

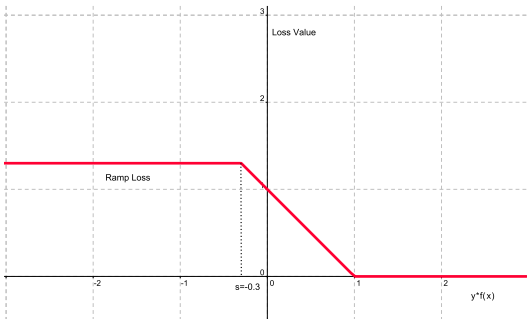

Fig. 5. Ramp Loss with s=-0.3

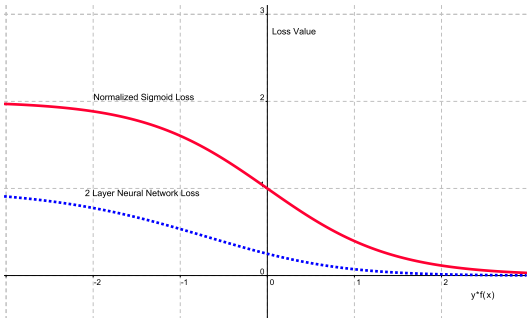

Fig. 6. Normalized Sigmoid Loss and 2 Layer Neural Network Loss

example. We believe that the outliers (misclassified examples far from the classifier) are getting more attention than they should because of the characteristics of convex loss functions. Prior research [20], [21] shows that it is possible to achieve an SVM classification algorithm where training errors are no longer support vectors. Different non-convex loss functions have been introduced.

1) Non-Smooth Non-convex Loss Functions: Mason et al. [22] proposed heuristic Direct Optimization Of Margins(DOOM) algorithm based on *sigmoid loss function* (see Figure 4),

$$V(f(\mathbf{x}), y)$$

$$= \begin{cases} (1.2-\gamma) - \gamma y f(\mathbf{x}) & \text{if } -1 \le y f(\mathbf{x}) \le 0 \\ (1.2-\gamma) - \frac{(1.2-2\gamma) y f(\mathbf{x})}{\theta} & \text{if } 0 < y f(\mathbf{x}) \le \theta \\ \gamma/(1-\theta) - \frac{\lambda y f(\mathbf{x})}{(1-\theta)} & \text{if } \theta < y f(\mathbf{x}) \le 1 \end{cases} \quad (12)$$

where $\theta \in (0,1)$, and γ was fixed at 0.1, and θ plays the role of a complexity parameter.

Shen et al. [12] proposed a non-convex loss function (see Figure 4).

$$V(f(\mathbf{x}), y) = \begin{cases} 1 - y f(\mathbf{x}) & \text{if } 0 \le y f(\mathbf{x}) \le 1 \\ 1 - Sign(y f(\mathbf{x})) & \text{otherwise} \end{cases} \quad (13)$$

The corresponding algorithm is called "ϕ-learning", which choose the initial guess obtained from either an SVM or a stochastic search. Moreover, multiple starting values are adopted to prevent the algorithm from being trapped with a local optimizer.

Ramp Loss (see Figure 5) is proposed by Collobert et al. [20], [21].

$$V(f(\mathbf{x}), y) = R_s(z) + R_s(-z) + const. \quad (14)$$

where $-1 < s \le 0$ is a hyper-parameter to be chosen and $R_s = \min(1-s, \max(0, 1-yf(\mathbf{x})))$. *Concave-Convex Procedure* (CCCP) [24] is adopted as the global optimization method to solve the corresponding non-convex optimization problem,

In [23], an Iterative Re-Weighted Least Squares (IRWLS) procedure was proposed, through which the Support Vector Classification (SVC) solution can be obtained for any convex or non-convex loss function. However, IRWLS procedure does not guarantee that the global minimum is found, instead, it only guarantee a solution with less empirical error with respect to the non-convex loss function. In their experiment, non-convex Sigmoid loss function shows better performance than the conventional SVC convex loss functions.

This type of loss functions are neither differentiable nor continual (ϕ-loss), which makes the corresponding optimization problem not applicable by some efficient optimization methods, for example Quasi-Newton Methods.

2) Smooth Non-convex Loss Functions: In [25] a non-convex *normalized sigmoid cost (lost) function* (15) (see Figure 6) is introduced.

$$V(f(\mathbf{x}), y) = 1 - \tanh(\lambda y f(\mathbf{x})) \quad (15)$$

An algorithm DOOM II is proposed corresponding to the normalized sigmoid loss function.

A similar loss function is defined in 2-layer Neural Newworks [7] (see Figure 6):

$$V(f(\mathbf{x}), y) = (1 - \frac{1}{1+\exp(-y f(\mathbf{x}))})^2 \quad (16)$$

Krause and Singer [26] propose a quite similar loss function, Logistic difference loss function.

$$V(f(\mathbf{x}), y) = ln(1 + e^{-y f(\mathbf{x})}) - ln(1 + e^{-y f(\mathbf{x}) - \mu}) \quad (17)$$

The upper bound of logistic difference loss function is controlled by parameter μ.

Loss functions as (15), (16), and (17) are all differentiable. The drawback of these loss functions is that there will be penalty for all training examples no matter whether they have

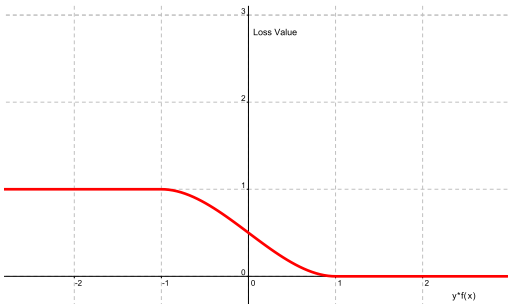

Fig. 7. A new non-convex loss function

been correctly classified or not. In other words, the value of these loss functions are always positive, and we cannot achieve 0 loss in any case, which can make the computation expensive and the corresponding estimation inaccurate.

III. Smoothed 0-1 Loss Function For Classification

According to the analysis of different convex and non-convex loss functions, we propose a new non-convex function, which we denote by *Smoothed 0-1 Loss Function*

$$V(t_i) = \begin{cases} 0 & t_i > 1 \\ \frac{1}{4}t_i^3 - \frac{3}{4}t_i + \frac{1}{2} & -1 \le t_i \le 1 \\ 1 & t_i < -1 \end{cases} \tag{18}$$

where $t_i = y_i(\mathbf{w} \cdot \mathbf{x}_i + b)$

A plot of this function is presented in Figure 7. The smoothed 0-1 loss function has certain desirable properties.

- It is bounded and nonconvex. For any data point $\mathbf{x}$, the proposed loss function are bounded between 0 and 1. The same as hinge loss and 0-1 loss, this loss function pay no penalty for data point $\mathbf{x}$ when $y * f(\mathbf{x}) > 1$. A classification model using this loss does not incur gain for pushing examples far from the decision boundary. When $y*f(\mathbf{x}) < -1$, it is the same as 0-1 loss, which makes the corresponding classification method robust for outliers. When $1 \le y * f(\mathbf{x}) \le -1$, the loss is decreasing. It is obvious that this loss function is a good approximation to 0-1 loss function with compelling justification.
- It is first order differentiable. Some efficient local optimization algorithms can be adopted to solve the corresponding classification model. Therefore we denote this loss function by Smoothed 0-1 Loss Function.

By incorporating smooth 0-1 loss function into Tikhonov regularization framework (1), we can have a new linear binary classification model.

$$\min_{\mathbf{w},b} \frac{1}{l}\sum_{i=1}^{l} V(t_i) + \lambda||\mathbf{w}||^2 \tag{19}$$

where $V(t_i)$ is defined by (18).

This is an unconstrained global optimization problem. To solve (19), global optimization method is required. Considering the loss function is strictly bounded between 0 and 1, the corresponding algorithm will be less sensitive to the choice of λ. Furthermore, the degree of improvement will become dramatic as the sample size increases, particular for noisy data sets with many outliers. Thus the corresponding algorithm will be more robust both with respect to the choice of parameter λ and with respect to noisy data sets.

It is obvious that the computational complexity of the non-convex loss based algorithms will be substantially higher than existing convex loss function based models, for example SVMs, we believe that the significant theoretical advantages will make it worthwhile to pursue further computational developments.

IV. A Method for Comparison of Different Loss Functions – the Quasisecant Method

Quasisecant Method (QSM) [27] is developed for solving the following unconstrained minimization problem:

$$\min f(x) \quad \text{subject to} \quad x \in \mathbb{R}^n \tag{20}$$

where the objective function f is assumed to be locally Lipschitz, but not necessarily differentiable or convex. For detailed definition of *quasisecant* of the function f please refer [27].

Algorithm 1 will be adopted as the main optimization algorithm for this paper. In this algorithm we denote a quasisecant set by $V_m(x_k)$.

Algorithm 1: The Quasisecant Method

Input: Start point $\mathbf{x}_0$, $\delta > 0$ and $c_1 \in (0, 1]$
Output: $\mathbf{x}^*$
Set k=0 ;
while $0_n \notin \partial f(x_k)$ **do**
  For given $\delta > 0$ and $c_1 \in (0, 1]$, compute the descent direction at $\mathbf{x} = \mathbf{x}_k$, we get the set $V_m(x_k)$ and an element v_k such that

$$||v_k||^2 = \min\{||v||^2 : v \in V_m(x_k)\}.$$

  Furthermore, either $||v^k|| \le \delta$ or for the search direction $g_k = -||v_k||^{-1}v^k$,

$$f(x_k + hg_k) - f(x_k) \le -c_1 h||v_k||.$$

  if $||v_k|| \le \delta$ **then**
    stop.
  end
  else
    Compute $x_{k+1} = x_k + \sigma_k g_k$, where σ_k is defined as follows

$$\sigma_k = \arg\max\{\sigma \ge 0 : f(x_k + \sigma g_k) - f(x_k) \le -c_2\sigma||v_k||\}.$$

    Set $k = k + 1$
  end
end

Numerical experiments conducted by Bagirov and Ganjehlou [27] have demonstrated that the QSM algorithm performs well when the objective function is nonsmooth, nonconvex. Another important feature of QSM algorithm is that this algorithm can be applied to both smooth and non-smooth functions, which make it possible to compare the performance of different loss functions under the framework of Tikhonov regularization model (1). In this paper, we adopt QSM algorithm to solve the optimization problem (19).

V. Binary Classification Algorithms

We have hypothesized that classifier designed with the smoothed 0-1 loss function (18) should be more robust than convex loss functions, for example the square loss function (3), and the hinge loss function (5). Moreover computational complexity of the classification algorithm based on smoothed 0-1 loss function should be better than non-smooth loss functions, such as the ramp loss function (14), and more accurate than smooth non-convex loss functions, for example the normalized sigmoid loss function (15). To test this, a set of binary classification algorithms are developed.

A. Algorithm for Convex Loss Function Based Binary Classification

For convex loss function based classification algorithm, the selection of the start point $(\mathbf{w}_0, b_0)$ is not important. Based on the Quasisecant algorithm, we propose an algorithm for convex loss function based binary classification. In this algorithm we set $(\mathbf{w}_0, b_0) = \mathbf{0}_n$.

Algorithm 2: Algorithm for Convex Loss Function Based Binary Classification

Input: Training set, Λ
Set start point $(\mathbf{w}_0, b_0) = \mathbf{0}_n$;
foreach $\lambda \in \Lambda$ **do**
  Apply Algorithm 1 for the computation of the optimization problem (1), where $V(f(\mathbf{x}), y)$ are corresponding convex loss function ;
  Denote the optimal solution by $(\mathbf{w}^*, b^*)$;
end

B. Algorithm for Non-Convex Loss Function Based Binary Classification

A good start point $(\mathbf{w}_0, b_0)$ is vital for algorithm for non-convex loss function based binary classification. In this paper, we select start point by algorithm 2 with hinge loss for the following reasons: (i) it is computational expensive to select a list of fixed start points; (ii) random start points are also computational expensive, moreover it is not reliable and not comparable for different results with random start points.

Algorithm 3: Algorithm for Non-Convex Loss Function Based Binary Classification

Input: Training set, Λ
Calculate a classifier $(\mathbf{w}_0, b_0)$ by Hinge Loss as a start point ;
foreach $\lambda \in \Lambda$ **do**
  Apply Algorithm 1 for the computation of the optimization problem (1), where $V(f(\mathbf{x}), y)$ are corresponding non-convex loss function ;
  Denote the optimal solution by $(\mathbf{w}^*, b^*)$;
end

TABLE I
Characteristics of the used data sets.

Data set	Number of Instances	Number of Attributes
liver-disorders	345	6
diabetes	768	8
heart	270	13
australian	690	14
vote	435	16

VI. Numerical Experiments

In this section, we investigate the classification performance of different loss functions in the framework of Tikhonov regularization. These loss functions are: smoothed 0-1 loss function (18), hinge loss function (5), square loss function (3), ramp loss function (14), and normalized sigmoid loss function (15).

For illustration, we make experiment comparison on the following benchmark data sets: liver-disorders, diabetes, heart disease, Australian, and vote (see Table I) from UCI repository [28]. Considering that there are only a small number of data points provided in these binary data sets, we randomly break a data set into ten equal sized subsets, and then considering the 10 train-test splits obtained by taking nine of the subsets as training and the remaining one as test set.

To evaluate the effectiveness, classification accuracy are used as performance evaluation metrics. λ are selected from a list of points evenly distributed in $[0, 2]$. Figure 8 illustrates the effectiveness of the smoothed 0-1 loss function based algorithm on liver data set. It is easy to find that Tikhonov regularization parameter λ plays an important role in controlling the overfitting problem. When λ is small, the calculated classifier can fit the training data very well, with over 74% accuracy. However, this classifier performs very bad on the test set. By increasing λ, the training accuracy decreases sharply, while the test accuracy increases accordingly. The experiment on the ramp loss function based classification algorithm shows similar trend (see Figure 9). The reason is that the classifier $\mathbf{w}$ has been smoothed by increasing λ and therefore less overfitting the training set. With the λ keeps increasing, the training and test accuracy decrease simultaneously when the $\mathbf{w}$ dominate the empirical risk model.

We need point out here that because the Liver-disorder

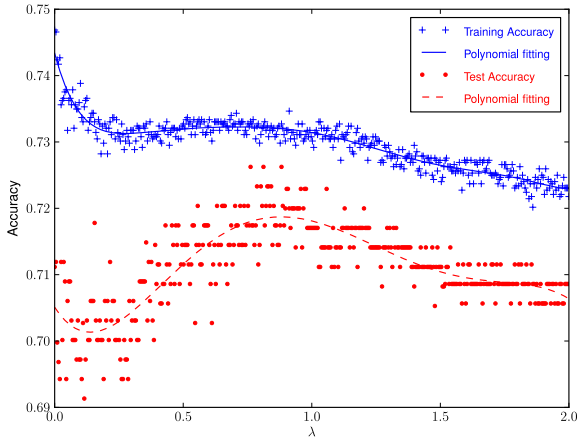

Fig. 8. Training and test accuracy of Liver-disorder data set calculated by Smoothed 0-1 Loss Function Based Binary Classification Algorithm

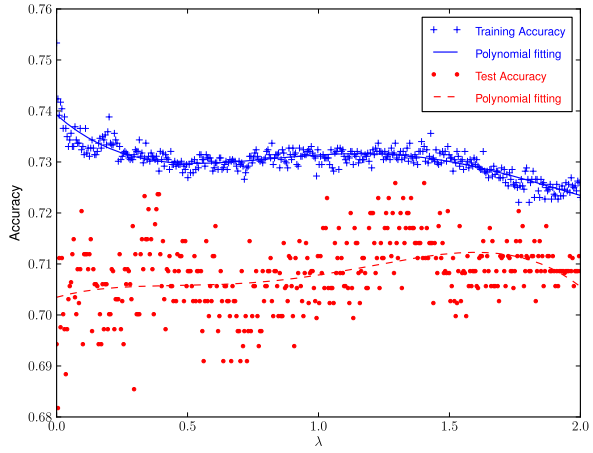

Fig. 9. Training and test accuracy of Liver-disorder data set calculated by Ramp Loss Function Based Binary Classification Algorithm

data set is a highly noisy data set, the test accuracy are not stable with respect of λ. For low noisy data sets, for example, Australian data set (see Figure 10), the training and test accuracy are quite consistent with the selection of λ. Moreover, because of the low noisy condition, best prediction accuracy is achieved when regularization parameter is quite small ($\lambda = 0.005$).

The average results of the 10 fold train-test splits are reported in Table II. From this comparison, we can see that nonconvex loss function based algorithms can achieve better generalization accuracy compared to convex loss function based algorithms. The overall training accuracy of nonconvex loss function based algorithms are better than convex loss function based algorithms, which shows that the nonconvex loss function can fit training examples better.

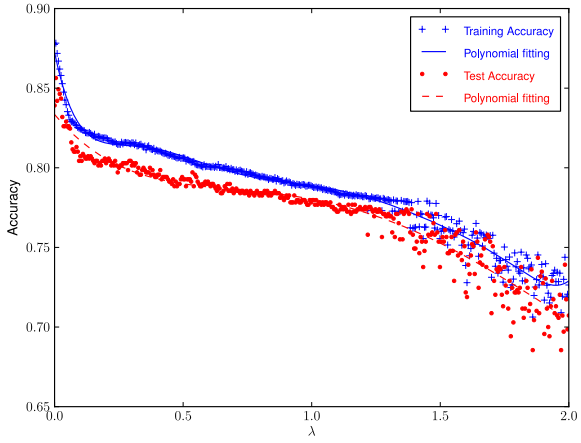

Fig. 10. Training and test accuracy of Australian data set calculated by Smoothed 0-1 Loss Function Based Binary Classification Algorithm

TABLE III
COMPARISON OF AVERAGE TRAINING TIME OVER 20 RUNS (SEC.)

Data Set	Convex		NonConvex		
	Square	Hinge	Smoothed 0-1	Ramp	Normalized Sigmoid
liver	**0.031**	0.437	**0.531**	0.812	0.625
diabetes	**0.047**	2.593	**2.858**	3.281	3.578
heart	**0.062**	5.047	**5.188**	5.703	5.250
australian	**0.046**	5.187	**5.562**	6.921	5.891
vote	**0.047**	1.109	2.250	**2.078**	2.843

Particularly, the smoothed 0-1 loss function can achieve better generalization accuracy on noisy data sets, for example liver-disorders and diabetes, which support our hypothesis in Section III.

Table III presents the average CPU time (in seconds) over 20 runs for one overall training phase (e.g. data loading, error computing, and optimization). As expected, among five classification algorithms, square loss function based algorithm is the fastest one. Except for the vote data set, smoothed 0-1 loss function based algorithm is the quickest among the nonconvex loss function (e.g. ramp and normalized sigmoid loss functions) based algorithms.

VII. CONCLUSION

Based on the analysis of some popular loss functions with respect to their convexity, continuity and differentiability, we propose a smoothed 0-1 loss function for binary classification. The new proposed loss function has some desirable properties: such as (i) it is an ideal approximation of 0-1 loss function; (ii) it is first order derivative. On the bases of Quasisecant method, we employ two binary classification algorithms for convex and non-convex loss functions, and compare the performance of different loss functions. We conduct experiments on several binary data sets for binary classification from UCI repository. The results show that non-convex loss function outperforms convex loss functions and our new proposed smoothed 0-1 loss function works well on noisy data sets.

TABLE II
COMPARISON OF CLASSIFICATION ACCURACY FOR BOTH TRAINING AND TEST SETS

	Smoothed 0-1 Loss Smooth, Nonconvex		Hinge Loss Nonsmooth, Convex		Square Loss Smooth, Convex		Ramp Loss Nonsmooth, Nonconvex		Normalized Sigmoid Loss Smooth, Nonconvex	
	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)
liver	73.34%	**72.62%**	71.08%	71.45%	70.60%	69.90%	72.98%	**72.62%**	**73.40%**	**72.62%**
diabetes	**78.90%**	**77.98%**	77.18%	77.06%	77.33%	77.22%	77.42%	76.66%	78.79%	76.92%
heart	85.93%	84.07%	85.76%	84.44%	85.68%	**85.19%**	85.51%	83.70%	**86.91%**	83.70%
australian	**88.71%**	86.23%	87.21%	85.65%	87.65%	**87.39%**	86.31%	85.51%	87.79%	85.36%
vote	96.76%	**94.23%**	94.46%	**94.23%**	94.97%	94.21%	94.56%	**94.23%**	**97.34%**	**94.23%**

Future studies should include develop corresponding theories such as generalization and convergence rate. We will also extend the smoothed 0-1 loss function based binary classification algorithm to multiclass or even multilabel data classification, and perform experiments on large data sets. Efficient global optimization methods that suitable for the smoothed 0-1 loss function should also be explored extensively.

ACKNOWLEDGMENTS

The authors gratefully thank Adil Bagirov for his advice and support, particularly for his excellent Quasisecant optimization method and corresponding Fortran source codes.

REFERENCES

[1] B. Scholkopf and A.J. Smola. *Learning with kernels*. Citeseer, 2002.
[2] T. Evgeniou, T. Poggio, M. Pontil, and A. Verri. Regularization and statistical learning theory for data analysis. *Computational Statistics and Data Analysis*, 38(4):421–432, 2002.
[3] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.
[4] G. Wahba. *Spline models for observational data*. Society for Industrial Mathematics, 1990.
[5] T. Evgeniou, M. Pontil, and T. Poggio. Statistical learning theory: A primer. *International Journal of Computer Vision*, 38(1):9–13, 2000.
[6] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning. 2001.
[7] F. Li and Y. Yang. A loss function analysis for classification methods in text categorization. In *Machine Learning-International Workshop Then Conference*, volume 20, page 472, 2003.
[8] L. Rosasco, E.D. Vito, A. Caponnetto, M. Piana, and A. Verri. Are loss functions all the same? *Neural Computation*, 16(5):1063–1076, 2004.
[9] R.M. Rifkin. *Everything Old Is New Again: A Fresh Look at*. PhD thesis, Massachusetts Institute of Technology, 2002.
[10] O.L. Mangasarian and E.W. Wild. Proximal support vector machine classifiers. *Proceedings KDD-2001: Knowledge Discovery and Data Mining*, pages 77–86, 2001.
[11] J.D.M. Rennie and N. Srebro. Loss functions for preference levels: Regression with discrete ordered labels. In *Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling*, pages 180–186. Citeseer, 2005.
[12] X. Shen, G.C. Tseng, X. Zhang, and W.H. Wong. On [Psi]-Learning. *Journal of the American Statistical Association*, 98(463):724–735, 2003.
[13] T. Zhang and F.J. Oles. Text categorization based on regularized linear classification methods. *Information Retrieval*, 4(1):5–31, 2001.
[14] Yoav Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.
[15] J. Friedman, T. Hastie, and R. Tibshirani. Special invited paper. additive logistic regression: A statistical view of boosting. *Annals of statistics*, 28(2):337–374, 2000.
[16] T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32(1):56–85, 2004.
[17] C.C. Aggarwal and P.S. Yu. Outlier detection for high dimensional data. *ACM Sigmod Record*, 30(2):37–46, 2001.
[18] C.E. Brodley and M.A. Friedl. Identifying and eliminating mislabeled training instances. In *the 1996 13th National Conference on Artificial Intelligence, AAAI 96. Part 1(of 2)*, pages 799–805, 1996.
[19] L. Xu, K. Crammer, and D. Schuurmans. Robust support vector machine training via convex outlier ablation. In *the National Conference On Artificial Intelligence*, volume 21, page 536. AAAI Press, 2006.
[20] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive SVMs. *the Journal of Machine Learning Research*, 7:1687–1712, 2006.
[21] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Trading convexity for scalability. In *Proceedings of the 23rd international conference on Machine learning*, pages 179–208. ACM, 2006.
[22] L. Mason, P.L. Bartlett, and J. Baxter. Improved generalization through explicit optimization of margins. *Machine Learning*, 38(3):243–255, 2000.
[23] F. Perez-Cruz, A. Navia-Vazquez, A.R. Figueiras-Vidal, and A. Artes-Rodriguez. Empirical risk minimization for support vector classifiers. *IEEE Transactions on Neural Networks*, 14(2):296–303, 2003.
[24] AL Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.
[25] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent in function space. In *Proc. NIPS*, volume 12, pages 512–518. Citeseer, 1999.
[26] N. Krause and Y. Singer. Leveraging the margin more carefully. In *Proceedings of the twenty-first international conference on Machine learning*, page 63. ACM, 2004.
[27] A.M. Bagirov and A.N. Ganjehlou. A quasisecant method for minimizing nonsmooth functions. *Optimization Methods and Software*, 25(1):3–18, 2010.
[28] A. Asuncion and D. Newman. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, 2007. *URL http://www. ics. uci. edu/mlearn/MLRepository. html*.