Journal of Machine Learning Research 12 (2011) 1-55 Submitted 12/11; Published 12/11

t-Logistic Regression

Nan Ding DING10@PURDUE.EDU
Department of Computer Science
Purdue University
West Lafayette, IN 47907-2066, USA

S.V. N. Vishwanathan VISHY@STAT.PURDUE.EDU
Departments of Statistics and Computer Science
Purdue University
West Lafayette, IN 47907-2066, USA

Manfred Warmuth MANFRED@CSE.UCSC.EDU
Department of Computer Science and Engineering
University of California, Santa Cruz
Santa Cruz, CA, USA

Vasil Denchev VDENCHEV@PURDUE.EDU
Department of Computer Science
Purdue University
West Lafayette, IN 47907-2066, USA

Editor: U.N. Known

Abstract

We extend logistic regression by using t-exponential families which were introduced recently in statistical physics. We examine our algorithm for both binary classfication and multiclass classfication with both L_1 and L_2 regularizer. The objective function of our algorithm is non-convex, an efficient block coordinate descent optimization scheme is derived for estimating the parameters. Because of the nature of the loss function, our algorithm is tolerant to label noise. We examine our algorithm in a bunch of synthetic as well as real datasets.

1. Introduction

Many machine learning algorithms minimize a regularized risk (Teo et al., 2010):

$$J(\boldsymbol{\theta}) = \Omega(\boldsymbol{\theta}) + \mathrm{R}_{\text{emp}}(\boldsymbol{\theta}), \text{ where } \mathrm{R}_{\text{emp}}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} l(\mathbf{x}_i, y_i, \boldsymbol{\theta}). \tag{1}$$

Here, Ω is a regularizer which penalizes complex $\boldsymbol{\theta}$; and R_{emp}, the empirical risk, is obtained by averaging the loss l over the training dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$. The features of a data point $\mathbf{x}$ is extracted via a feature map Φ. In binary classification, the label is usually predicted via $\operatorname{sign}(\langle \Phi(\mathbf{x}), \boldsymbol{\theta} \rangle)$; while in multiclass classification, the label is predicted via $\max_c \{\langle \Phi(\mathbf{x}), \boldsymbol{\theta}_c \rangle\}$ where $\boldsymbol{\theta}_c$ denotes the subvector of $\boldsymbol{\theta}$ corresponds to the parameter of class c.

For a long period, convex losses have been a strong favorate by people, mainly because it ensures that the regularized risk minimization problem has a unique global optimum (Boyd and Vandenberghe, 2004) and the rate of convergence of the algorithm can be analytically obtained. However,

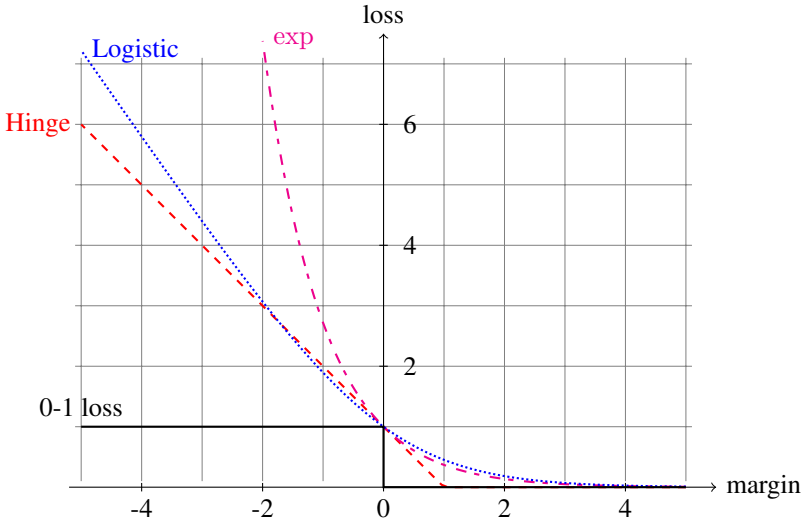


Figure 1: Some commonly used loss functions for binary classification. The 0-1 loss is non-convex. The hinge, exponential, and logistic losses are convex upper bounds of the 0-1 loss.

as was recently shown by Long and Servedio (2010), learning algorithms based on convex loss functions for binary classification are not robust[1] to noise[2]. In the binary classification, if we define the *margin* of a training example $(\mathbf{x}, y)$ as $u(\mathbf{x}, y, \boldsymbol{\theta}) := y \langle \phi(\mathbf{x}), \boldsymbol{\theta} \rangle$, then many popular loss functions for binary classification can be written as functions of the margin. Examples include[3]

$$l(u) = \max(0, 1 - u) \qquad \text{(Hinge Loss)} \tag{2}$$

$$l(u) = \exp(-u) \qquad \text{(Exponential Loss)} \tag{3}$$

$$l(u) = \log(1 + \exp(-u)) \qquad \text{(Logistic Loss).} \tag{4}$$

Intuitively, the convex loss functions grows at least linearly with slope $|l'(0)|$ as $u \in (-\infty, 0)$, which introduces the overwhelming impact from the data with $u \ll 0$. There has been some recent and some not-so-recent work on using non-convex loss functions Freund (2009) to alleviate the above problem.

In this paper, we continue this line of inquiry and propose the non-convex loss function which is firmly grounded in probability theory. By extending logistic regression from the exponential family to the t-exponential family, a natural extension of exponential family of distributions studied in statistical physics (Naudts, 2002, 2004a,b,c; Tsallis, 1988), we obtain the t-logistic regression algorithm. Furthermore, the binary t-logistic regression can be extended to multiclass t-logistic regression. We also show that L_1 and L_2 regularizer in t-logistic regression corresponds to the Student's t-distribution and a newly proposed t-Laplace distribution in the t-exponential family. We

1. There is no unique definition of robustness. For example, one of the definitions is through the outlier-proneness O'hagan (1979): $p(\boldsymbol{\theta} \,|\, \mathbf{X}, \mathbf{Y}, \mathbf{x}_{n+1}, y_{n+1}) \to p(\boldsymbol{\theta} \,|\, \mathbf{X}, \mathbf{Y})$ as $\mathbf{x}_{n+1} \to \infty$.
2. Although, the analysis of Long and Servedio (2010) is carried out in the context of boosting, we believe, the results hold for a larger class of algorithms which minimize a regularized risk with a convex loss function.
3. We slightly abuse notation and use $l(u)$ to denote $l(u(\mathbf{x}, y, \boldsymbol{\theta}))$.

give a simple block coordinate descent scheme that can be used to solve the resultant regularized risk minimization problem. Analysis of this procedure also intuitively explains why t-logistic regression is able to handle label noise.

Our paper is structured as follows: In section 2 we briefly review generalized exponential family, which includes Student's t-distribution and t-Laplace distribution. In section 3, we review the logistic regression in the exponential family both for binary classfication and multiclass classification. We then proposed t-logistic regression algorithm in section 6. In section **??** we utilize ideas from convex multiplicative programming to design an optimization strategy and give its convergence analysis. Experiments that compare our new approach to existing algorithms on a number of publicly available datasets are reported in section 9. It follows a discussion with the related research work in section **??** and the outlook in section 11.

2. Generalized exponential family of distributions

2.1 ϕ-exponential family of distribution

To define the generalized exponential family of distribution, we first have to review the generalizations of the $\log$ and $\exp$ functions which were introduced in statistical physics (Naudts, 2002, 2004a,b,c). Some extensions and machine learning applications were presented in (Sears, 2008).

The ϕ-logarithm function is defined in the domain of $\mathbb{R}^+$ as

$$\log_\phi(x) = \int_1^x \frac{1}{\phi(u)} du \tag{5}$$

where $\phi(u) > 0$ is an increasing function defined on $\mathbb{R}^+$. The ϕ-exponential function $\exp_\phi$ is defined as the inverse function of the ϕ-logarithm function. Both $\log_\phi$ and $\exp_\phi$ function generalizes the usual $\exp$ and $\log$ function, which is recovered as $\phi(u) = u$. Many familiar properties of $\exp$ are therefore preserved. $\exp_\phi$ function is a non-negative, convex and monotonically increasing function passing $(0, 1)$ and $\log_\phi$ function is concave, monotonically increasing and passing $(1, 0)$. Besides, it is easy to verify that the first derivative of $\log_\phi$ and $\exp_\phi$ are,

$$\frac{d\left(\log_\phi(x)\right)}{dx} = \frac{1}{\phi(x)}\,, \frac{d\left(\exp_\phi(x)\right)}{dx} = \phi(\exp_\phi(x)) \tag{6}$$

However, one key property missing is that $\log_\phi(ab) \neq \log_\phi(a) + \log_\phi(b)$ and $\exp_\phi(a+b) \neq \exp_\phi(a)\exp_\phi(b)$ when $\phi(u) \neq u$.

Analogous to the exponential family of distributions, the ϕ-exponential family of distributions is defined as (Naudts, 2004c; Sears, 2008):

$$p(\mathbf{x};\boldsymbol{\theta}) := \exp_\phi\left(\langle\phi(\mathbf{x}), \boldsymbol{\theta}\rangle - g_\phi(\boldsymbol{\theta})\right). \tag{7}$$

where g_ϕ is the log-partition function ensures that $p(\mathbf{x};\boldsymbol{\theta})$ is normalized. In general, unlike $g(\boldsymbol{\theta})$ in the exponential family, there is no closed form solution exists for computing $g_\phi(\boldsymbol{\theta})$ exactly. However, the following theorem, which was first proved by Naudts in (Naudts, 2004c), shows that $g_\phi(\boldsymbol{\theta})$ still preserves a few important properties.

Theorem 2.1 *$g_\phi(\boldsymbol{\theta})$ is a strictly convex function. In addition, if the following regularity condition*

$$\int \nabla_\theta p(x;\theta)dx = \nabla_\theta \int = \nabla_\theta 1 = 0 p(x;\theta)dx \tag{8}$$

holds, then

$$\nabla_{\boldsymbol{\theta}} g_\phi(\boldsymbol{\theta}) = \mathbb{E}_{q_\phi(\mathbf{x};\boldsymbol{\theta})}\left[\phi(\mathbf{x})\right]. \tag{9}$$

where $q_\phi(\mathbf{x};\boldsymbol{\theta})$ *is the* escort distribution

$$q_\phi(x;\boldsymbol{\theta}) := \phi(p(x;\boldsymbol{\theta}))/Z(\boldsymbol{\theta}) \tag{10}$$

where $Z(\boldsymbol{\theta}) = \int \phi(p(x;\boldsymbol{\theta}))dx$.

Proof To prove convexity, we rely on the elementary arguments. Recall that $\exp_\phi$ is an increasing and strictly convex function. Choose θ_1 and θ_2 such that $g_\phi(\theta_i) < \infty$ for $i = 1, 2$, and let $\alpha \in (0, 1)$. Set $\theta_\alpha = \alpha\theta_1 + (1-\alpha)\theta_2$, and observe that

$$\begin{aligned}
&\int \exp_\phi\left(\langle\phi(x), \theta_\alpha\rangle - \alpha g_\phi(\theta_1) - (1-\alpha)g_\phi(\theta_2)\right) dx \\
&< \alpha \int \exp_\phi(\langle\phi(x), \theta_1\rangle - g_\phi(\theta_1))dx + (1-\alpha)\int \exp_\phi(\langle\phi(x), \theta_2\rangle - g_\phi(\theta_2))dx = 1.
\end{aligned}$$

On the other hand, we also have

$$\int \exp_\phi\left(\langle\phi(x), \theta_\alpha\rangle - g_\phi(\theta_\alpha)\right) dx = 1.$$

Again, using the fact that $\exp_t$ is an increasing function, we can conclude from the above two equations that

$$g_\phi(\theta_\alpha) < \alpha g_\phi(\theta_1) + (1-\alpha)g_\phi(\theta_2).$$

This shows that g_ϕ is a strictly convex function.

To show (9), use (8) and combining with the fact that $\frac{d}{du}\exp_\phi(u) = \phi(\exp_\phi(u))$, to write

$$\begin{aligned}
\int \nabla_\theta p(x;\theta)dx &= \int \nabla_\theta \exp_\phi\left(\langle\phi(x), \theta\rangle - g_\phi(\theta)\right) dx \\
&= \int \phi\left(\exp_\phi\left(\langle\phi(x), \theta\rangle - g_\phi(\theta)\right)\right)\left(\phi(x) - \nabla_\theta g_\phi(\theta)\right)dx \\
&\propto \int q_\phi(x;\theta)(\phi(x) - \nabla_\theta g_\phi(\theta))dx = 0.
\end{aligned}$$

Rearranging terms and using $\int q_\phi(x;\theta)dx = 1$ directly yields (9). ■

Therefore, the main difference from $\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta})$ of the exponential family is that $\nabla_{\boldsymbol{\theta}} g_\phi(\boldsymbol{\theta})$ is equal to the expectation of its escort distribution $q_\phi(x;\boldsymbol{\theta})$ instead of $p(x;\boldsymbol{\theta})$.

2.2 t-exponential family of distribution

One example of the ϕ-exponential family which draws us particular attention is the t-exponential family. The t-exponential/logarithm function as well as the t-exponential family was first proposed in 1980s by Tsallis (where he called it q-exponential family, but we tend to use t instead of q to avoid confusion with the escort distribution q).

The $\exp_t$ and $\log_t$ function is a special case of the $\exp_\phi$ and $\log_\phi$ function with $\phi(u) = u^t$ for $t > 0$:

$$\exp_t(x) := \begin{cases} \exp(x) & \text{if } t = 1 \\ [1 + (1-t)x]_+^{1/(1-t)} & \text{otherwise.} \end{cases} \tag{11}$$

where $(\cdot)_+ = \max(\cdot, 0)$. Some examples are shown in Figure 2. The inverse of $\exp_t$ namely $\log_t$ is

$$\log_t(x) := \begin{cases} \log(x) & \text{if } t = 1 \\ \left(x^{1-t} - 1\right)/(1-t) & \text{otherwise.} \end{cases} \tag{12}$$

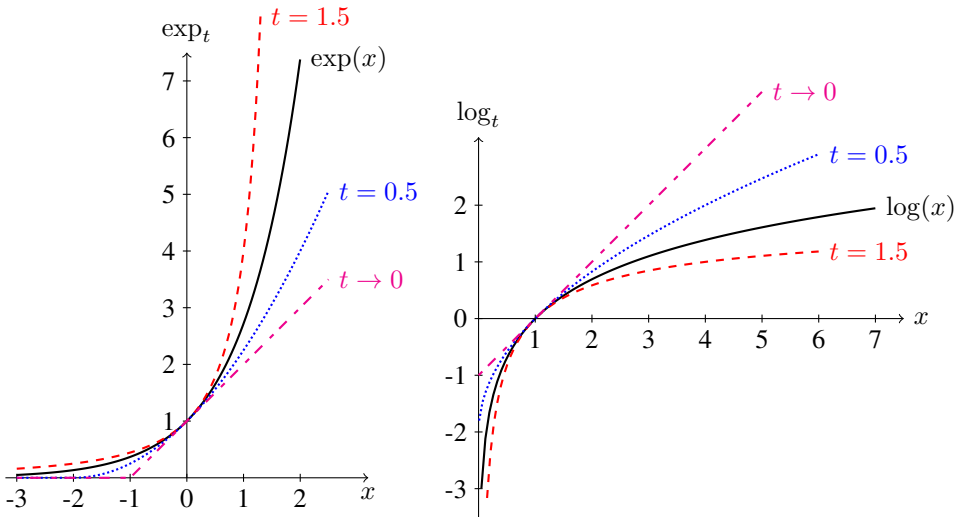


Figure 2: Left: $\exp_t$ and Middle: $\log_t$ for various values of t indicated. The right figure depicts the t-logistic loss functions for different values of t. When $t = 1$, we recover the logistic loss

From Figure 2, you can see that $\exp_t$ decays towards 0 more slowly than the $\exp$ function for $t > 1$. This important property leads to a family of heavy tailed distributions.

3. Logistic Regression

Logistic regression is a discriminative model which is mainly used in classifications. In particular, we are given a labeled dataset $(\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$. The feature $\mathbf{x}_i$'s are drawn from some d-dimensional domain $\mathcal{X}$. The labels y_i being categorical and C is the number of classes. Given a family of conditional distributions parametrized by $\boldsymbol{\theta}$, making a standard i.i.d. assumption about the data allows us to write,

$$p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) = \prod_{i=1}^{m} p(y_i | \, \mathbf{x}_i, \boldsymbol{\theta}) \tag{13}$$

Logistic regression computes a maximum likelihood (ML) estimate for $\boldsymbol{\theta}$ by minimizing

$$-\log p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) = -\sum_{i=1}^{m} \log p(y_i | \, \mathbf{x}_i; \boldsymbol{\theta}) \tag{14}$$

as a function of $\boldsymbol{\theta}$. To avoid overfitting to the data, it is common to add a regularizer of the parameter $\boldsymbol{\theta}$, which yields the maximum a-posteriori (MAP) estimate of $\boldsymbol{\theta}$,

$$-\log p(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{X}) = -\sum_{i=1}^{m} \log p(y_i | \, \mathbf{x}_i; \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}) + \text{const.} \tag{15}$$

where $p(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{X}) = p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) / p(\mathbf{y} \mid \mathbf{X})$ follows the Bayes rule and $p(\mathbf{y} \mid \mathbf{X})$ is a constant of $\boldsymbol{\theta}$.

The regularizer is usually assumed to be a zero mean isotropic Gaussian prior, which yields:

$$-\log p(\boldsymbol{\theta}) = \frac{\lambda}{2} \left\| \boldsymbol{\theta} \right\|_2^2 + \text{ const. } . \tag{16}$$

This is also called an L_2 regularizer. On the other hand, when the feature space is huge, L_1 regularizer is favored because it enforces sparsity on the components of $\boldsymbol{\theta}$. To this end, Laplace prior is normally chosen as the prior, which yields

$$-\log p(\boldsymbol{\theta}) = \lambda \left\| \boldsymbol{\theta} \right\|_1 + \text{ const. } . \tag{17}$$

In the logistic regression, $p(y_i | \, \mathbf{x}_i; \boldsymbol{\theta})$ is modeled as the conditional exponential family of distributions

$$p(y_i | \, \mathbf{x}_i; \boldsymbol{\theta}) = \exp\left(\langle \Phi(\mathbf{x}_i, y_i), \boldsymbol{\theta} \rangle - g(\boldsymbol{\theta} \mid \mathbf{x}_i) \right), \tag{18}$$

We will concentrate on the linear classifier, this allows us to simplify,

$$\Phi(\mathbf{x}, k) = \left(\underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{1, \ldots, y-1}, \mathbf{x}, \underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{y+1, \ldots, C} \right). \tag{19}$$

with $\mathbf{0}$ the d-dimensional all-zero vector. Therefore,

$$\langle \Phi(\mathbf{x}_i, y_i), \boldsymbol{\theta} \rangle = \boldsymbol{\theta}_{y_i}^T \mathbf{x}_i$$

$\boldsymbol{\theta}$ is a $d \times C$-dim vector,

$$\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_C) \, . \tag{20}$$

and $\boldsymbol{\theta}_c$ is a d-dim vector which is the c-th segment of $\boldsymbol{\theta}$.

$$p(y_i | \, \mathbf{x}_i; \boldsymbol{\theta}) = \exp\left(\boldsymbol{\theta}_{y_i}^T \mathbf{x}_i - g(\boldsymbol{\theta} \mid \mathbf{x}_i) \right) \tag{21}$$

The log-partition function

$$g(\boldsymbol{\theta} \mid \mathbf{x}_i) = \log \left(\sum_{c=1}^{C} \exp\left(\boldsymbol{\theta}_c^T \mathbf{x}_i \right) \right). \tag{22}$$

4. Logistic Loss for Binary Classification

For a binary classification problem, we have

$$p(y_i|\,\mathbf{x}_i;\boldsymbol{\theta}) = \exp\left(\boldsymbol{\theta}_{y_i}^T\,\mathbf{x}_i - g(\boldsymbol{\theta}\,|\,\mathbf{x}_i)\right), \tag{23}$$

where

$$g(\boldsymbol{\theta}\,|\,\mathbf{x}_i) = \log\left(\exp\left(\boldsymbol{\theta}_1^T\,\mathbf{x}_i\right) + \exp\left(\boldsymbol{\theta}_2^T\,\mathbf{x}_i\right)\right). \tag{24}$$

If we rewrite $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2$, and slightly abuse notation by using $y_i \in \{+1, -1\}$ to represent $y_i = \{1, 2\}$, then through a simple modification, we can equivalently write,

$$p(y_i|\,\mathbf{x}_i;\tilde{\boldsymbol{\theta}}) = \exp\left(\frac{1}{2}y_i\tilde{\boldsymbol{\theta}}^T\,\mathbf{x}_i - \tilde{g}(\tilde{\boldsymbol{\theta}}|\,\mathbf{x}_i)\right) \tag{25}$$

where

$$\tilde{g}(\tilde{\boldsymbol{\theta}}|\,\mathbf{x}_i) = \log\left(\exp\left(\frac{1}{2}\tilde{\boldsymbol{\theta}}^T\,\mathbf{x}_i\right) + \exp\left(-\frac{1}{2}\tilde{\boldsymbol{\theta}}^T\,\mathbf{x}_i\right)\right). \tag{26}$$

and

$$-\log p(y_i|\,\mathbf{x}_i;\tilde{\boldsymbol{\theta}}) = -\frac{1}{2}y_i\tilde{\boldsymbol{\theta}}^T\,\mathbf{x}_i + \log\left(\exp(\frac{1}{2}\tilde{\boldsymbol{\theta}}^T\,\mathbf{x}_i) + \exp(-\frac{1}{2}\tilde{\boldsymbol{\theta}}^T\,\mathbf{x}_i)\right) \tag{27}$$

$$= \log\left(1 + \exp(-y_i\tilde{\boldsymbol{\theta}}^T\,\mathbf{x}_i)\right) \tag{28}$$

Binary logistic regression, also is called the logistic loss, has widely been used as a convex surrogate loss for binary classification. For a long period, convex losses $l(\mathbf{x}, y, \tilde{\boldsymbol{\theta}})$ have been a strong favorite by people, mainly because it ensures that the empirical risk of $\{(\mathbf{x}_i, y_i)\}$ with $i = 1, \ldots, m$

$$\sum_{i=1}^{m} l(\mathbf{x}_i, y_i, \tilde{\boldsymbol{\theta}})$$

has a unique global optimum (Boyd and Vandenberghe, 2004) and the rate of convergence of the algorithm can be analytically obtained. If we define the *margin* of a training example $(\mathbf{x}, y)$ as $u(\mathbf{x}, y, \tilde{\boldsymbol{\theta}}) := y\left\langle\tilde{\boldsymbol{\theta}}, \mathbf{x}\right\rangle$, then many popular loss functions for binary classification can be written as functions of the margin. Examples include:

$$l(u) = \max(0, 1 - u) \qquad \text{(Hinge Loss)} \tag{29}$$

$$l(u) = \exp(-u) \qquad \text{(Exponential Loss)} \tag{30}$$

$$l(u) = \log(1 + \exp(-u)) \qquad \text{(Logistic Loss).} \tag{31}$$

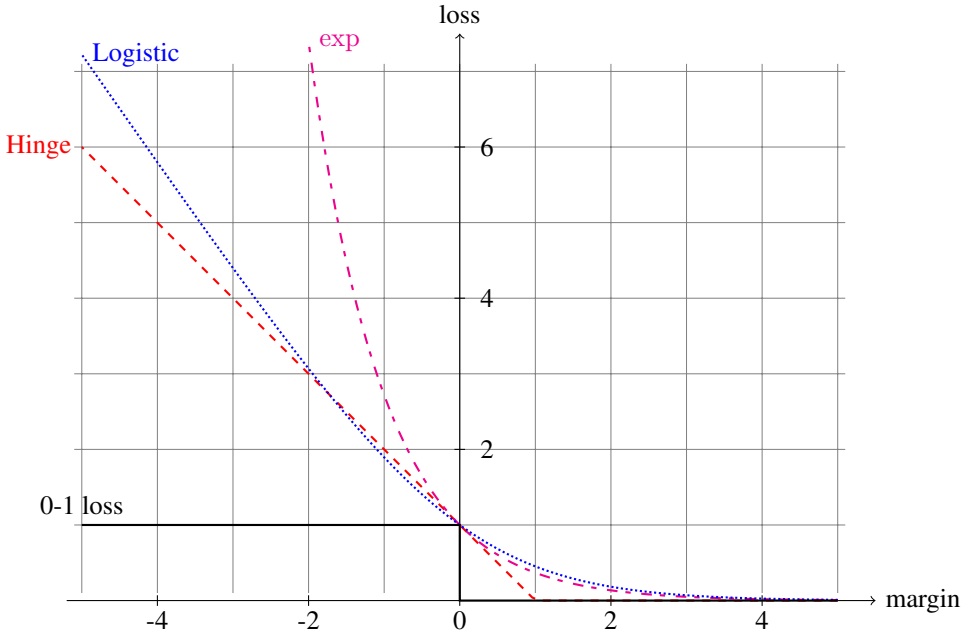


Figure 3: Some commonly used loss functions for binary classification. The 0-1 loss is non-convex. The hinge, exponential, and logistic losses are convex upper bounds of the 0-1 loss.

4.1 Shortcomings of Convex Losses

However, as was recently shown by (Long and Servedio, 2010), learning algorithms based on convex loss functions for binary classification are not robust to noise[4]. In (Long and Servedio, 2010), Long and Servedio constructed the following dataset to show that minimizing a convex losses are not tolerant to label noise (the label noise is added by flipping a portion of the labels of the training data). Each data point has a 21-dimension feature vector and play one of three possible roles: **large margin examples** (25%, $x_{1,2,\ldots,21} = y$); **pullers** (25%, $x_{1,\ldots,11} = y$, $x_{12,\ldots,21} = -y$); and **penalizers** (50%, Randomly select and set 5 of the first 11 coordinates and 6 out of the last 10 coordinates to y, and set the remaining coordinates to $-y$). They show that although using convex losses can classify the clean data perfectly, adding 10% label noise into the dataset can fool the convex classifiers.

This phenomenon is illustrated in Figure 5. The black double arrow is the true classifier. However, after adding 10% label noise, the convex classifier changes to the red double arrow, which is no longer able to distinguish the **penalizers** and leads to around 25% of error.

The reason is intuitively shown in Figure 4, the convex loss functions grows at least linearly with slope $|l'(0)|$ as $u \in (-\infty, 0)$, which introduces the overwhelming impact from the data with $u \ll 0$. Therefore, the true black classifier suffers from large penalties of the flipped large margin data and is beaten up by the red classifier even when the red one misclassifies more points.

4. Although, the analysis of Long and Servedio (2010) is carried out in the context of boosting, we believe, the results hold for a larger class of algorithms which minimize a regularized risk with a convex loss function.

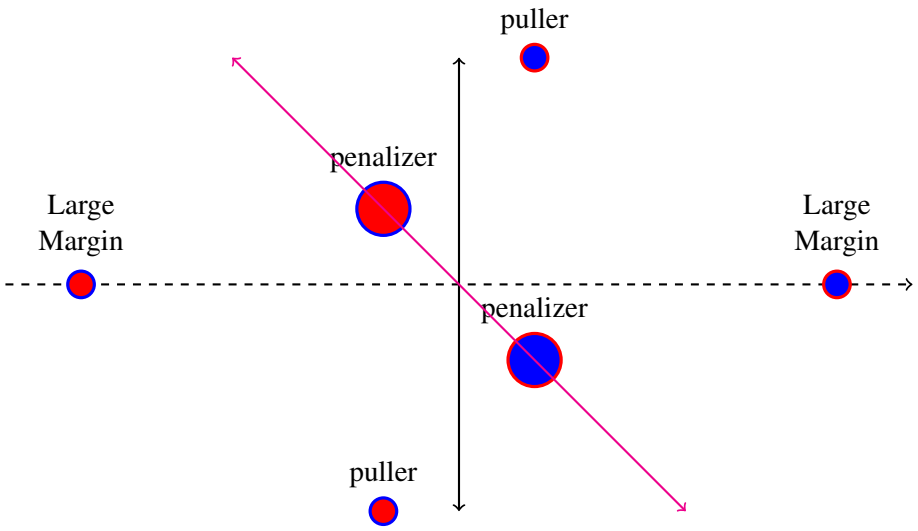


Figure 4: The Long-Servedio dataset

5. Optimal Condition and Robustness

We have seen in the last section that the logistic regression is not robust against label noise. In this section, we discuss the condition of robustness for the discriminative models. For the dataset $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, the maximum likelihood estimate (MLE) $\boldsymbol{\theta}^*$ of a discriminative model has to satisfy,

$$\begin{aligned} 0 &= -\frac{1}{m}\frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}^*) \\ &= -\frac{1}{m}\sum_{i=1}^{m} \frac{\partial}{\partial \boldsymbol{\theta}} \log p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}^*) \end{aligned}$$

We consider a model to be robust, if $\boldsymbol{\theta}^*$ is the MLE of the dataset $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, with $m \to \infty$, then after adding any new data point $(\mathbf{x}_{m+1}, y_{m+1})$, the optimal condition still holds for $\boldsymbol{\theta}^*$, that is

$$\lim_{m\to\infty} \frac{1}{m+1}\sum_{i=1}^{m} \frac{\partial}{\partial \boldsymbol{\theta}} \log p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}^*) + \frac{1}{m+1}\frac{\partial}{\partial \boldsymbol{\theta}} \log p(y_{m+1} \mid \mathbf{x}_{m+1}, \boldsymbol{\theta}^*) = 0$$

Since that

$$\lim_{m\to\infty} \frac{1}{m+1}\sum_{i=1}^{m} \frac{\partial}{\partial \boldsymbol{\theta}} \log p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}^*) = 0$$

it is equivalent to that for any data point $(\mathbf{x}_{m+1}, y_{m+1})$,

$$\lim_{m\to\infty} \frac{1}{m+1}\frac{\partial}{\partial \boldsymbol{\theta}} \log p(y_{m+1} \mid \mathbf{x}_{m+1}, \boldsymbol{\theta}^*) = 0$$

Furthermore, because the dataset $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ are chosen arbitrarily, $\boldsymbol{\theta}^*$ can be any parameter matrix except the trivial all 0 matrix; besides the additional data point $(\mathbf{x}_{m+1}, y_{m+1})$ can also be chosen arbitrarily. Because of the existence of the regularizer, we assume that $\boldsymbol{\theta}^*$ is bounded by a constant as $|\,\boldsymbol{\theta}^*\,|_\infty \leq M$.

Overall, we define the robustness as follows,

Definition 5.1 *A model is robust if and only if,* $\forall\, \mathbf{x}_i, y_i$ *and* $\forall\, \boldsymbol{\theta} \setminus \mathbf{0}^{C\times d}$ *such that* $|\,\boldsymbol{\theta}^*\,|_\infty \leq M$*,*

$$-\frac{\partial}{\partial\, \boldsymbol{\theta}} \log p(y_i|\, \mathbf{x}_i, \boldsymbol{\theta})$$

is bounded.

Additionally, the following lemma, which first shown in Liu (2004), is a straightforward result from Definition 5.1,

Lemma 5.2 *A model is robust if and only if,* $\forall\, \mathbf{x}_i, y_i$ *and* $\forall\, \boldsymbol{\theta} \setminus \mathbf{0}^{C\times D}$ *such that* $|\,\boldsymbol{\theta}\,|_\infty \leq M$*,*

$$I(\mathbf{x}_i, y_i, \boldsymbol{\theta}) = \left\langle \boldsymbol{\theta}, -\frac{\partial}{\partial\, \boldsymbol{\theta}} \log p(y_i|\, \mathbf{x}_i, \boldsymbol{\theta}) \right\rangle$$

is bounded.

5.1 Optimal Condition and Robustness of Logistic regression

For logistic regression, the optimal condition is

$$\begin{aligned} 0 &= -\frac{1}{m}\frac{\partial}{\partial\, \boldsymbol{\theta}} \log p(\mathbf{y}\,|\,\mathbf{X}, \boldsymbol{\theta}^*) \\ &= -\frac{1}{m}\sum_{i=1}^{m} \frac{\partial}{\partial\, \boldsymbol{\theta}} \left(\langle \Phi(\mathbf{x}_i, y_i), \boldsymbol{\theta}\rangle - g(\boldsymbol{\theta}\,|\,\mathbf{x}_i)\right) \\ &= -\frac{1}{m}\sum_{i=1}^{m} (\Phi(\mathbf{x}_i, y_i) - \mathbb{E}_{p_i^*}[\Phi(\mathbf{x}_i, y)]) \end{aligned}$$

which yields,

$$\sum_{i=1}^{m} \Phi(\mathbf{x}_i, y_i) = \sum_{i=1}^{m} \mathbb{E}_{p_i^*}[\Phi(\mathbf{x}_i, y)] \tag{32}$$

for any class c,

$$\sum_{i=1}^{m} \Phi(\mathbf{x}_i)\delta(y_i = c) = \sum_{i=1}^{m} \Phi(\mathbf{x}_i) p_i^*(c) \tag{33}$$

where p_i^* denotes $p(y|\, \mathbf{x}_i, \boldsymbol{\theta}^*)$ for brevity. This optimal condition is sometimes referred as moment matching in the exponential family.

Now we justify the robustness of logistic regression,

$$\begin{aligned} I(\mathbf{x}_i, y_i, \boldsymbol{\theta}) &= \left\langle \boldsymbol{\theta}, -\frac{\partial}{\partial\, \boldsymbol{\theta}} \log p(y_i|\, \mathbf{x}_i, \boldsymbol{\theta}) \right\rangle \\ &= (\Phi(\mathbf{x}_i, y_i) - \mathbb{E}_{p_i}[\Phi(\mathbf{x}_i, y)])^T\, \boldsymbol{\theta} \end{aligned}$$

where p_i denote for $p(y_i|\,\mathbf{x}_i, \boldsymbol{\theta})$ for simplicity. Apparently, under the following condition, $I(\mathbf{x}_i, y_i, \boldsymbol{\theta})$ is unbounded.

- $p(c|\,\mathbf{x}_i, \boldsymbol{\theta}) = \max_y\{p(y|\,\mathbf{x}_i, \boldsymbol{\theta})\}$,
- $y_i \neq c$. This implies $\boldsymbol{\theta}_c^T\,\mathbf{x}_i \geq \boldsymbol{\theta}_{y_i}^T\,\mathbf{x}_i$,
- $\mathbf{u}_i = \boldsymbol{\theta}_c^T\,\mathbf{x}_i \to \infty$.

Therefore, we justified that the logistic regression is not robust.

6. t-logistic Regression

From the last section, we observe that designing a robust discriminative model should come to the following two rules:

- Given dataset $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, $\boldsymbol{\theta}^*$ satisfies the optimal condition

$$-\frac{1}{m}\sum_{i=1}^{m}\frac{\partial}{\partial\,\boldsymbol{\theta}}\log p(y_i|\,\mathbf{x}_i, \boldsymbol{\theta}^*) = 0 \tag{34}$$

- Given any data point $(\mathbf{x}_i, y_i)$ and $\forall\,\boldsymbol{\theta}\,\backslash\mathbf{0}^{C\times d}$ such that $|\,\boldsymbol{\theta}\,|_\infty \leq M$,

$$\left\langle \boldsymbol{\theta}, -\frac{\partial}{\partial\,\boldsymbol{\theta}}\log p(y_i|\,\mathbf{x}_i, \boldsymbol{\theta})\right\rangle \tag{35}$$

  is bounded.

We first introduce our discriminative model which satisfy the above two rules, and then will justify them in the following subsections. Our solution is surprisingly simple: we generalize the logistic regression by replacing the conditional exponential family distribution by conditional t-exponential family distribution ($t > 1$):

$$p(y|\,\mathbf{x}, \boldsymbol{\theta}) = \exp_t\left(\langle\Phi(\mathbf{x}, y), \boldsymbol{\theta}\rangle - g_t(\boldsymbol{\theta}\,|\,\mathbf{x})\right) \tag{36}$$

$$= \exp_t\left(\boldsymbol{\theta}_y^T\,\mathbf{x} - g_t(\boldsymbol{\theta}\,|\,\mathbf{x})\right) \tag{37}$$

where the log-partition function g_t satisfies

$$\sum_{c=1}^{C}\exp_t\left(\boldsymbol{\theta}_c^T\,\mathbf{x} - g_t(\boldsymbol{\theta}\,|\,\mathbf{x})\right) = 1. \tag{38}$$

Since no closed form solution to g_t of the above equation exists in general, the efficiency of the t-logistic regression is highly dependent on the computational efficiency of the $g_t(\boldsymbol{\theta}\,|\,\mathbf{x})$. In the following, we show how $g_t(\boldsymbol{\theta}\,|\,\mathbf{x})$ can be computed numerically.

6.1 Numerical Estimation of g_t

The basic idea of solving g_t numerically is to take a iterative scheme. First, let us denote $u_c = \boldsymbol{\theta}_c^T \mathbf{x}$, so that (38) is simplified as

$$\sum_{c=1}^{C} \exp_t(u_c - g_t(\mathbf{u})) = 1 \tag{39}$$

It is clear that there exists $\tilde{\mathbf{u}}$ and a function

$$Z(\tilde{\mathbf{u}}) = \sum_{c=1}^{C} \exp_t(\tilde{\mathbf{u}}) \,. \tag{40}$$

which satisfy,

$$\begin{aligned} \exp_t(u_c - g_t(\mathbf{u})) &= \frac{1}{Z(\tilde{\mathbf{u}})} \exp_t(\tilde{u}_c) \,. \\ &= \exp_t\left(\underbrace{Z(\tilde{\mathbf{u}})^{t-1}\tilde{u}_c}_{=u_c} + \underbrace{\log_t(\frac{1}{Z(\tilde{\mathbf{u}})})}_{=-g_t(\mathbf{u})} \right) \quad \forall c \in \{1, \ldots, C\} \,. \end{aligned}$$

The last equation uses the definition of $\exp_t$ function. The iterative algorithm for computing $g_t(\mathbf{u})$ goes as follows:

1. Compute $u^* = \max\{\mathbf{u}\}$
2. Offset $\mathbf{v} = \mathbf{u} - u^*$
3. Initialize $\tilde{\mathbf{u}} = \mathbf{v}$
4. Compute $Z = \sum_{c=1}^{C} \exp_t(\tilde{\mathbf{u}})$
5. Update $\tilde{\mathbf{u}} = Z^{1-t}\mathbf{v}$
6. If not converge, go to step 4; otherwise go to step 7
7. Output $g_t = u^* - \log_t(1/Z)$

Our experiments show that $g_t(\mathbf{u})$ can be obtained with high accuracy in less than 10 iterations. To illustrate, we let $C \in \{10, 20, \ldots, 100\}$ and we randomly generate $\mathbf{u} \in [-10, 10]^C$, and compute the corresponding $g_t(\mathbf{u})$. We compare the time spent in estimating $g_t(\mathbf{u})$ by the iterative scheme and by calling Matlab `fsolve` function averaged over 100 runs using Matlab 7.1 in a 2.93 GHz Dual-Core CPU. And we present the results in Table 1. The results show that our iterative method scales well with C the number of classes, thus making it efficient enough for problems involving a large number of classes.

Table 1: Average time (in milliseconds) spent by our iterative scheme and `fsolve` on solving $g_t(\mathbf{u})$.

C	10	20	30	40	50	60	70	80	90	100
`fsolve`	8.1	8.3	8.1	8.7	9.6	9.8	10.0	10.2	10.3	10.7
iterative	0.3	0.3	0.3	0.4	0.4	0.4	0.3	0.3	0.4	0.5

6.2 Optimal Condition and Robustness of t-logistic Regression

The MLE of the t-logistic regression $\boldsymbol{\theta}^*$ is,

$$\begin{aligned}
0 &= -\frac{1}{m}\frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{y} \,|\, \mathbf{X}, \boldsymbol{\theta}^*) \\
&= -\frac{1}{m}\sum_{i=1}^{m} \frac{\partial}{\partial \boldsymbol{\theta}} \log \exp_t \left(\boldsymbol{\theta}^*_{y_i} \mathbf{x}_i - g_t(\boldsymbol{\theta}^* \,|\, \mathbf{x}_i) \right) \\
&= -\frac{1}{m}\sum_{i=1}^{m} (\Phi(\mathbf{x}_i, y_i) - \mathbb{E}_{q_i^*}[\Phi(\mathbf{x}_i, y)]) p(y_i | \, \mathbf{x}_i, \boldsymbol{\theta}^*)^{t-1}
\end{aligned}$$

And the optimal condition for the t-logistic regression is,

$$\sum_{i=1}^{m} \Phi(\mathbf{x}_i, y_i)\xi_i = \sum_{i=1}^{m} \mathbb{E}_{q_i^*}[\Phi(\mathbf{x}_i, y)]\xi_i \tag{41}$$

for any class c,

$$\sum_{i=1}^{m} \xi_i \Phi(\mathbf{x}_i)\delta(y_i = c) = \sum_{i=1}^{m} \xi_i \Phi(\mathbf{x}_i) q_i^*(c) \tag{42}$$

where $q_i^*(y) \propto p_i^{*t}(y)$ is the escort distribution of $p_i^*(y) = p(y_i | \, \mathbf{x}_i, \boldsymbol{\theta}^*)$ and

$$\xi_i = p(y_i | \, \mathbf{x}_i, \boldsymbol{\theta}^*)^{t-1}. \tag{43}$$

is the weight of each data point.

There are two remarks about this optimal condition:

Firstly, it might be a bit weird to see q_i^* instead of p_i^*. As a matter of fact, q_i^* simply amplifies the distribution of p_i^* to make it closer to $\mathbf{e}_{c_i^*}$, where

$$c_i^* = \operatorname*{argmax}_c \langle \boldsymbol{\theta}_c^*, \mathbf{x}_i \rangle .$$

Note that if we use $\mathbf{e}_{c_i^*}$ to replace $\mathbb{E}_{p_i^*}[y]$ in the optimal condition of the logistic regression, it yields to the perception method, where

$$\sum_{i=1}^{m} \Phi(\mathbf{x}_i, y_i) = \sum_{i=1}^{m} \Phi(\mathbf{x}_i, c_i^*)$$

Secondly, when $t > 1$, the impact of the i-th data point is controlled by its weight ξ_i. Intuitively, these weights associated with each of the data points will dampen the contribution of any outliers since they always have small ξ_i. However, do these extra weights make the t-logistic regression robust? We justify it as follows.

According to (36) and Lemma 5.2, we have

$$\begin{aligned}
\log p(y_i|\,\mathbf{x}_i,\boldsymbol{\theta}) &= \log(\exp_t(\langle\Phi(\mathbf{x}_i,y_i),\boldsymbol{\theta}\rangle - g_t(\boldsymbol{\theta}\,|\,\mathbf{x}_i))) \\
I(\mathbf{x}_i,y_i,\boldsymbol{\theta}) &= -\exp_t^{t-1}(\langle\Phi(\mathbf{x}_i,y_i),\boldsymbol{\theta}\rangle - g_t(\boldsymbol{\theta}\,|\,\mathbf{x}_i))\left\langle\boldsymbol{\theta}, \frac{\partial}{\partial\boldsymbol{\theta}}(\langle\Phi(\mathbf{x}_i,y_i),\boldsymbol{\theta}\rangle - g_t(\boldsymbol{\theta}\,|\,\mathbf{x}_i))\right\rangle \\
&= -\exp_t^{t-1}(\langle\Phi(\mathbf{x}_i,y_i),\boldsymbol{\theta}\rangle - g_t(\boldsymbol{\theta}\,|\,\mathbf{x}_i))(\langle\Phi(\mathbf{x}_i,y_i) - \mathbb{E}_{q_i}[\Phi(\mathbf{x}_i,y)],\boldsymbol{\theta}\rangle) \\
&= \frac{\langle\mathbb{E}_{q_i}[\Phi(\mathbf{x}_i,y_i)],\boldsymbol{\theta}\rangle - \langle\Phi(\mathbf{x}_i,y_i),\boldsymbol{\theta}\rangle}{1+(t-1)(g_t(\boldsymbol{\theta}\,|\,\mathbf{x}_i) - \langle\Phi(\mathbf{x}_i,y_i),\boldsymbol{\theta}\rangle)}
\end{aligned}$$

For simplicity, let us use the vector $\mathbf{u}_i$, where each element $u_{iy} = \boldsymbol{\theta}_y^T\,\mathbf{x}_i$, and write

$$p(y_i|\,\mathbf{x}_i,\boldsymbol{\theta}) = p(y_i|\,\mathbf{u}_i) = \exp_t(u_{iy_i} - g_t(\mathbf{u}_i))$$

where

$$\sum_{c=1}^{C}\exp_t(\mathbf{u}_{ic} - g_t(\mathbf{u}_i)) = 1$$

and $\frac{\partial}{\partial\,\mathbf{u}_i}g_t(\mathbf{u}_i) = \mathbb{E}_{q_i}[\mathbf{e}_y] = [q_{i1};\ldots;q_{iC}]$ with $q_{iy} \propto p(y|\,\mathbf{u}_i)^t$.

Then we have

$$\begin{aligned}
I(\mathbf{x}_i,y_i,\boldsymbol{\theta}) &= I(\mathbf{u}_i,y_i) \\
&= -\exp_t^{t-1}(u_{iy_i} - g_t(\mathbf{u}_i))(\mathbf{u}_{iy_i} - \mathbb{E}_{q_i}[\mathbf{e}_y]^T\,\mathbf{u}_i) \\
&= \frac{\mathbb{E}_{q_i}[\mathbf{e}_y]^T\,\mathbf{u}_i - u_{iy_i}}{1+(t-1)(g_t(\mathbf{u}_i) - u_{iy_i})}
\end{aligned}$$

We now justify that as $u_{ic}\to\infty$ $(\forall c)$, $I(\mathbf{u}_i,y_i)$ is bounded.

Firstly, if $|\,\mathbb{E}_{q_i}[\mathbf{e}_y]^T\,\mathbf{u}_i - u_{iy_i}|$ is bounded as $u_{ic}\to\infty$ $(\forall c)$, then because $t > 1$ and $g_t(\mathbf{u}_i) - \mathbf{u}_{iy_i} \geq 0$,

$$\lim_{\mathbf{u}_i\to\infty}|I(\mathbf{u}_i,y_i)| \leq |\,\mathbb{E}_{q_i}[\mathbf{e}_y]^T\,\mathbf{u}_i - u_{iy_i}|$$

is also bounded.

On the other hand, if $\mathbb{E}_{q_i}[\mathbf{e}_y]^T\,\mathbf{u}_i - u_{iy_i}\to\infty$, then we will apply the L'hospital principle,

$$\begin{aligned}
&\lim_{u_{ic}\to\infty} I(\mathbf{u}_i,y_i)\quad(\forall c) \\
=&\lim_{u_{ic}\to\infty}\frac{\mathbb{E}_{q_i}[\mathbf{e}_y]^T\,\mathbf{u}_i - u_{iy_i}}{1+(t-1)(g_t(\mathbf{u}_i) - u_{iy_i})}\quad(\forall c) \\
=&\lim_{u_{ic}\to\infty}\frac{\frac{\partial}{\partial u_{ic}}\left(\mathbb{E}_{q_i}[\mathbf{e}_y]^T\,\mathbf{u}_i - u_{iy_i}\right)}{\frac{\partial}{\partial u_{ic}}\left(1+(t-1)(g_t(\mathbf{u}_i) - u_{iy_i})\right)}\quad(\forall c) \\
=&\frac{q_{ic} - \delta(y_i = c)}{(t-1)(q_{ic} - \delta(y_i = c))}\quad(\forall c) \\
=&\frac{1}{t-1}.
\end{aligned}$$

where the third equation comes because of the property that $\frac{\partial}{\partial u_{ic}} g_t(\mathbf{u}_i) = \mathbb{E}_{q_i}[y_c]$.

Therefore, we conclude that the t-logistic regression is robust because $I(\mathbf{x}_i, y_i, \boldsymbol{\theta})$ is bounded for any $(\mathbf{x}_i, y_i)$ and $\boldsymbol{\theta}$.

7. t-logistic Loss for Binary Classification

After a general discussion on the t-logistic regression, we now closely investigate the t-logistic regression for binary classification problem. The binary t-logistic regression is interesting because we are able to visualize its loss function and justify its Bayes consistency.

The t-logistic regression for binary classification is simply a two class example of the t-logistic regression,

$$p(y_i|\,\mathbf{x}_i;\boldsymbol{\theta}) = \exp_t\left(\boldsymbol{\theta}_{y_i}^T\,\mathbf{x}_i - g_t(\boldsymbol{\theta}\,|\,\mathbf{x}_i)\right), \tag{44}$$

where

$$\exp_t\left(\boldsymbol{\theta}_1^T\,\mathbf{x} - g_t(\boldsymbol{\theta}\,|\,\mathbf{x})\right) + \exp_t\left(\boldsymbol{\theta}_2^T\,\mathbf{x} - g_t(\boldsymbol{\theta}\,|\,\mathbf{x})\right) = 1. \tag{45}$$

Just like the logistic regression, the binary t-logistic regression can also be rewritten as $\tilde{\boldsymbol{\theta}} = \frac{1}{2}(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2)$, and $y \in \{+1, -1\}$,

$$p(y_i|\,\mathbf{x}_i;\tilde{\boldsymbol{\theta}}) = \exp_t\left(y_i\tilde{\boldsymbol{\theta}}^T\,\mathbf{x}_i - \tilde{g}_t(\tilde{\boldsymbol{\theta}}|\,\mathbf{x}_i)\right)$$

where,

$$\exp_t\left(\tilde{\boldsymbol{\theta}}^T\,\mathbf{x} - \tilde{g}_t(\tilde{\boldsymbol{\theta}}|\,\mathbf{x})\right) + \exp_t\left(-\tilde{\boldsymbol{\theta}}^T\,\mathbf{x} - \tilde{g}_t(\tilde{\boldsymbol{\theta}}|\,\mathbf{x})\right) = 1.$$

We call the negative log-likelihood of the t-logistic regression, the t-logistic loss. For a data point with margin u, the t-logistic loss is

$$l(u) = -\log\exp_t(u - \tilde{g}_t(u)) \qquad (t\text{-logistic loss})$$

where

$$\exp_t(u - \tilde{g}_t(u)) + \exp_t(-u - \tilde{g}_t(u)) = 1.$$

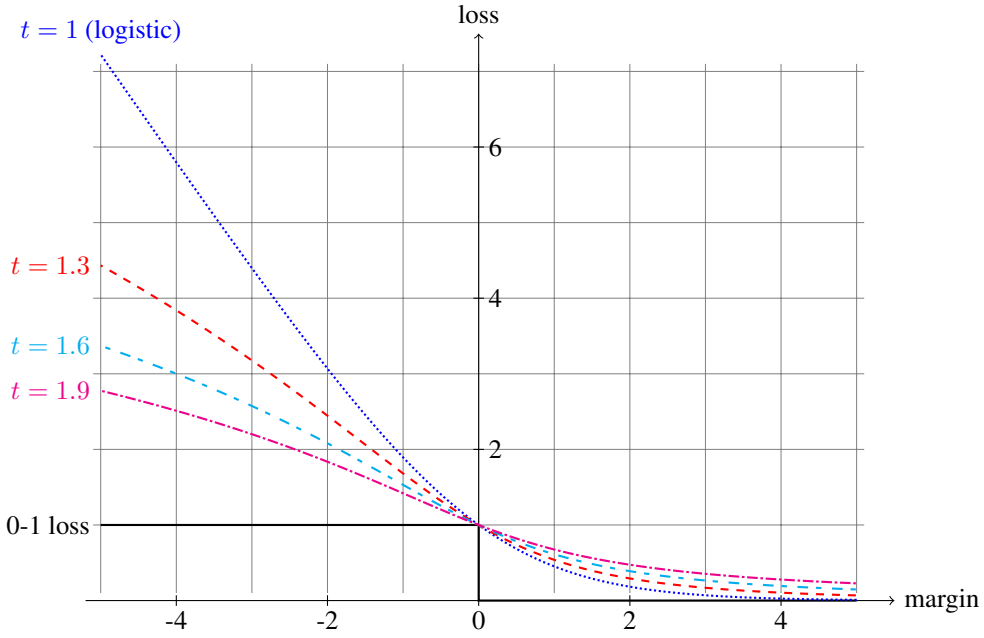
When $t = 2$, we obtain an analytical form for the binary t-logistic loss. To see this, note that when $t = 2$, $\exp_t(x) = 1/(1-x)$, therefore,

$$\frac{1}{1 - u + \tilde{\mathrm{g}}_t(u)} + \frac{1}{1 + u + \tilde{\mathrm{g}}_t(u)} = 1$$

which leads to $\tilde{\mathrm{g}}_t(u) = \sqrt{1+u^2}$ and

$$l(u) = \log(1 - u + \sqrt{1+u^2})$$

Notice that we no longer obtain a convex loss function. The loss function is curling down as the margin of a data point is too negative, which usually is the indication of an outlier. Therefore, the benefit of the nonconvexity is that the losses by those outliers are not as large as the convex losses, so that the model will be more robust against the outliers.



7.1 Bayes Consistency

It is well-known that all the convex losses mentioned earlier for binary classification are Bayes consistent. In this subsection, we want to justify the Bayes consistency of the t-logistic loss. We first review some concepts in (Bartlett et al., 2006). The binary classification algorithms are essentially proposed to optimize the empirical risks which are,

$$R_L(f) = \mathbb{E}_{\mathbf{x},y}[L(yf(\mathbf{x}))]$$

For any given $(\mathbf{x}, y)$, if we denote $\eta(\mathbf{x}) = p(y = 1|\,\mathbf{x})$, then the associated conditional risk given $\mathbf{x}$ can be written as,

$$C_L(\eta, f) = \mathbb{E}_{y|\,\mathbf{x}}[L(yf(\mathbf{x}))] = \eta L(f) + (1 - \eta)L(-f)$$

The 0-1 loss can be written as

$$L_{0/1}[yf(\mathbf{x})] = \begin{cases} 1, & yf(\mathbf{x}) < 0 \\ 0, & \text{otherwise} \end{cases}$$

then the minimization of the 0-1 loss gives,

$$\mathbb{E}_{y|\,\mathbf{x}}[L_{0/1}[yf(\mathbf{x})]] = \eta L_{0/1}[f^*(\mathbf{x})] + (1 - \eta)L_{0/1}[-f^*(\mathbf{x})] = L_{0/1}[(2\eta - 1)f^*(\mathbf{x})]$$

which yields to,

$$f^*(\mathbf{x}) = \text{sign}[2\eta(\mathbf{x}) - 1]$$

Definition 7.1 *A Bayes consistent loss function is the class of loss function, for which $f_L^*(\eta)$ the minimizer of $C_L(\eta, f)$ satisfies*

$$f_L^*(\eta) = \text{sign}[2\eta - 1] \tag{46}$$

For t-logistic regression, we have

$$L(f) = \log \exp_t(\frac{1}{2}yf - g_t(f)) \tag{47}$$

And

$$\begin{aligned} C_L(\eta, f) &= \eta L(f) + (1-\eta)L(-f) && (48)\\ &= \eta \log \exp_t(\frac{1}{2}f - g_t(f)) + (1-\eta)\log \exp_t(-\frac{1}{2}f - g_t(f)) && (49)\\ &= \eta \log \exp_t(\frac{1}{2}f - g_t(f)) + (1-\eta)\log(1 - \exp_t(\frac{1}{2}f - g_t(f))). && (50) \end{aligned}$$

Minimizing over f results in the f_L^* which satisfies $\eta = \exp_t(f^* - g_t(f^*))$, and therefore

$$\begin{aligned} f_L^*(\eta) &= \log_t \eta - \log_t(1-\eta) && (51)\\ C_L^*(\eta) &= -\eta \log \eta - (1-\eta)\log(1-\eta) && (52) \end{aligned}$$

It is then clear that t-logistic loss is Bayes consistent since it satisfies (46).

8. Generalization of the Logistic Regression by Trust Function

As we have shown in the last section, t-logistic regression provides an interesting generalization of the logistic regression. Meanwhile, there are also other different kinds of generalization of the logistic regression.

Recall that the logistic regression takes,

$$p(y|\,\mathbf{x};\boldsymbol{\theta}) = \exp\left(\langle \Phi(\mathbf{x}, y), \boldsymbol{\theta}\rangle - g(\boldsymbol{\theta}\,|\,\mathbf{x})\right), \tag{53}$$

Denoting the margin to be $\mathbf{u}$ where $u_c = \langle \Phi(\mathbf{x}, c), \boldsymbol{\theta}\rangle$, we propose a different generalization of the logistic regression by replacing the margin $\mathbf{u}$ with $s(\mathbf{u})$ where $s : \mathbb{R} \to \mathbb{R}$, based on the energy-based learning framework(). This function modifies the logistic regression as follows:

$$\begin{aligned} p(y|\,\mathbf{x};\boldsymbol{\theta}) &= \exp\left(s(\langle \Phi(\mathbf{x}, y), \boldsymbol{\theta}\rangle) - \tilde{g}(\boldsymbol{\theta}\,|\,\mathbf{x})\right) && (54)\\ &= \exp\left(s(u_y) - \tilde{g}(\mathbf{u})\right), && (55) \end{aligned}$$

where

$$\begin{aligned} \tilde{g}(\boldsymbol{\theta}\,|\,\mathbf{x}) &= \log\left(\sum_{c=1}^{C} \exp(s(\langle \Phi(\mathbf{x}, c), \boldsymbol{\theta}\rangle))\right) && (56)\\ &= \log\left(\sum_{c=1}^{C} \exp(s(u_c))\right) && (57) \end{aligned}$$

We call s an *s-function*, and its derivative s' a *trust function* if they have the following properties:

1. s antisymmetric: $s(-u) = -s(u)$, $s(0) = 0$.
2. $s(\infty) = \infty$ which will assure that the loss remains unbounded.

3. $s'(u) \in [0, 1]$ and $s'(0) = 1$ because we want the trust at $s'(u)$ of the linear activation u be a probability.
4. s' symmetric: $s'(-u) = s'(u)$.
5. The trust $s'(u)$ is decreasing in the interval $[0..\infty]$ i.e. as we go away from zero.
6. $\text{limit}_{u\to\infty}\, u\, s'(u)$ is increasing in u for $u \geq 0$ and asympotes at a constant.

Our main family of trust functions will be scaled versions of the $\text{arcsinh}(u)$ function[5] (See Figure **??**):

$$s_q(u) := \frac{1}{q}\,\text{arcsinh}(q\,u), \quad s'_q(u) = \frac{1}{\sqrt{q^2\,u^2+1}},$$
$$\text{and } \lim_{u\to\infty}\, u\, s'_q(u) = \frac{1}{q},$$

where $q > 0$ is a non-negative scaling parameter. Note that $\text{limit}_{q\to 0}\, s_q(u) = u$ and therefore we define $s_0(u)$ as the identity function u. The function $s_0(u)$ the straight line in Figure **??**, and as q increases, $s_q(u)$ bends to the right above zero and to the left below zero.

The corresponding trust functions $s_q(u)$ are depicted in Figure **??**. Note that the $s'_0(u) = 1$, ie. full trust everywhere. However, the larger q the less trust is given to linear activations with high absolute value.

8.1 Binary Classification

For binary classification, by using the label $y = \{+1, -1\}$ and denoting $u = y\tilde{\boldsymbol{\theta}}^T\,\mathbf{x}$, one can generalize the logistic loss to obtain the *trust loss* by,

$$l(u) = \log(1 + \exp(s(-u)))$$

Especially, when $s(u) = \text{arcsinh}(u)$, we have,

$$l(u) = \log(1 - u + \sqrt{1+u^2})$$

It is clear to see that the trust loss is equal to the t-logistic loss as $t = 2$.

8.2 Optimal Condition and Robustness

For the Trust Regression, the optimal condition is

$$\begin{aligned} 0 &= -\frac{1}{m}\frac{\partial}{\partial\,\boldsymbol{\theta}}\log p(\mathbf{y}\,|\,\mathbf{X}, \boldsymbol{\theta}^*) \\ &= -\frac{1}{m}\sum_{i=1}^{m}\frac{\partial}{\partial\,\boldsymbol{\theta}}\left(s(\langle\Phi(\mathbf{x}_i, y_i), \boldsymbol{\theta}\rangle) - \tilde{\mathrm{g}}(\boldsymbol{\theta}\,|\,\mathbf{x}_i)\right) \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left(s'(\langle\Phi(\mathbf{x}_i, y_i), \boldsymbol{\theta}\rangle)\Phi(\mathbf{x}_i, y_i) - \mathbb{E}_{p_i^*}[s'(\langle\Phi(\mathbf{x}_i, y), \boldsymbol{\theta}\rangle)\Phi(\mathbf{x}_i, y)]\right) \end{aligned}$$

5. $\text{arcsinh}(u) = \ln(u + \sqrt{1+u^2})$

In particular, for any class c, we have

$$\begin{aligned}
0 &= -\frac{1}{m}\sum_{i=1}^{m}\left(s'(\langle\Phi(\mathbf{x}_i, y_i), \boldsymbol{\theta}\rangle)\Phi(\mathbf{x}_i)\delta(y_i = c) - s'(\langle\Phi(\mathbf{x}_i, c), \boldsymbol{\theta}\rangle)\Phi(\mathbf{x}_i)p_i^*(c)\right) \\
&= -\frac{1}{m}\sum_{i=1}^{m}\left(s'(\langle\Phi(\mathbf{x}_i, c), \boldsymbol{\theta}\rangle)\Phi(\mathbf{x}_i)\delta(y_i = c) - s'(\langle\Phi(\mathbf{x}_i, c), \boldsymbol{\theta}\rangle)\Phi(\mathbf{x}_i)p_i^*(c)\right) \\
&= -\frac{1}{m}\sum_{i=1}^{m}\tilde{\xi}_{ic}\Phi(\mathbf{x}_i)\left(\delta(y_i = c) - p_i^*(c)\right)
\end{aligned}$$

which immediately yields to the optimal condition,

$$\sum_{i=1}^{m}\tilde{\xi}_{ic}\Phi(\mathbf{x}_i)\delta(y_i = c) = \sum_{i=1}^{m}\tilde{\xi}_{ic}\Phi(\mathbf{x}_i)p_i^*(c). \tag{58}$$

where $p_i^*(y)$ denotes $p(y|\,\mathbf{x}_i, \boldsymbol{\theta}^*)$ for brevity, and

$$\tilde{\xi}_{ic} = s'(\langle\Phi(\mathbf{x}_i, c), \boldsymbol{\theta}\rangle) \tag{59}$$

is the weight of each data. The second equation is because as $y_i \neq c$, the first term is always 0.

From the above gradient, tt can be justified the trust regression is robust.

$$\begin{aligned}
I(\mathbf{x}_i, y_i, \boldsymbol{\theta}) &= \left\langle \boldsymbol{\theta}, -\frac{\partial}{\partial\,\boldsymbol{\theta}}\log p(y_i|\,\mathbf{x}_i, \boldsymbol{\theta})\right\rangle \\
&= \left(s'(\langle\Phi(\mathbf{x}_i, y_i), \boldsymbol{\theta}\rangle)\Phi(\mathbf{x}_i, y_i) - \mathbb{E}_{p_i^*}[s'(\langle\Phi(\mathbf{x}_i, y), \boldsymbol{\theta}\rangle)\Phi(\mathbf{x}_i, y)]\right)^T\boldsymbol{\theta} \\
&= s'(\langle\Phi(\mathbf{x}_i, y_i), \boldsymbol{\theta}\rangle)\,\langle\Phi(\mathbf{x}_i, y_i), \boldsymbol{\theta}\rangle - \mathbb{E}_{p_i^*}[s'(\langle\Phi(\mathbf{x}_i, y), \boldsymbol{\theta}\rangle)\,\langle\Phi(\mathbf{x}_i, y), \boldsymbol{\theta}\rangle]
\end{aligned}$$

Denoting $u_{yi} = \langle\Phi(\mathbf{x}_i, y), \boldsymbol{\theta}\rangle$, and making use of the sixth propertyof the s-function that $s'(u)u$ is upper bounded by a constant, we have

$$I(\mathbf{x}_i, y_i, \boldsymbol{\theta}) = s'(u_{y_i i})u_{y_i i} - \mathbb{E}_{p_i^*}[s'(u_{yi})u_{yi}]$$

is bounded.

8.3 t-logistic Regression vs. Trust Regression

We have already seen the the t-logistic regression and the trust regression are closely related in the binary classification settings. However, it is easy to verify that these two algorithms are completely different in general. Since both t-logistic regression and trust regression are generalizations of the logistic regression. A natural question to ask is which one is better?

To answer this question, let us first look at the gradient of these log-likelihood functions. For any class c, the gradient of the logistic regression is

$$-\frac{\partial}{\partial\,\boldsymbol{\theta}_c}\log p(\mathbf{y}\,|\,\mathbf{X}, \boldsymbol{\theta}) = -\sum_{i=1}^{m}\Phi(\mathbf{x}_i)\left(\delta(y_i = c) - p_i(c)\right)$$

The gradient of the t-logistic regression is,

$$-\frac{\partial}{\partial \boldsymbol{\theta}_c} \log p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) = -\sum_{i=1}^{m} \xi_i \Phi(\mathbf{x}_i) \left(\delta(y_i = c) - q_i(c)\right)$$

where $\xi_i = p_i(y_i)^{t-1}$. The gradient of the trust regression is,

$$-\frac{\partial}{\partial \boldsymbol{\theta}_c} \log p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) = -\sum_{i=1}^{m} \tilde{\xi}_{ic} \Phi(\mathbf{x}_i) \left(\delta(y_i = c) - p_i(c)\right)$$

where $\tilde{\xi}_{ic} = s'(\langle \Phi(\mathbf{x}_i, c), \boldsymbol{\theta} \rangle)$. Basically, the contribution of each data point on the gradients, can be decomposed to the product of two terms. The first is the *weight* term, which is equal to 1, ξ_i and $\tilde{\xi}_{ic}$ for logistic regression, t-logistic regression, and trust regression respectively; and the second is *sign* term, which is larger than 0 as $c = y_i$ and less than 0 otherwise.

The sign terms of the three algorithms are very similar to each other. The main difference of the above three algorithms lies in the weights of each data point. The weight terms essentially measure the goodness of the data points. In the t-logistic regression, ξ_i is dependent on both $\mathbf{x}_i$ and y_i; on the other hand, $\tilde{\xi}_{ic}$ in the trust regression is independent of the label y_i of the data. In supervised learning, the goodness of a data point $(\mathbf{x}_i, y_i)$ is not only dependent on its feature $\mathbf{x}_i$, but also dependent on its label y_i. In other words, for the t-logistic regression, if there is an outlier with low $p_i(y_i)$, then it has low impacts on all $\boldsymbol{\theta}_c$'s. For the trust regression, since the weights across different classes c's are different, the outliers may still have impact for some $\boldsymbol{\theta}_c$'s.

We will compare the logistic regression, the t-logistic regression, as well as the trust regression in the experiments.

9. Experimental Evaluation

In the experiment, we evaluate the t-logistic regression in synthetic datasets, real datasets for binary classification, and real datasets for multiclass classification.

9.1 Synthetic Dataset

We first examine the t-logistic regression on the following two benchmark synthetic datasets: Long-Servedio and Mease-Wyner. The Long-Servedio dataset is an artificially constructed dataset to show that algorithms which minimize a differentiable convex loss are not tolerant to label noise Long and Servedio (2010). The Mease-Wyner is another synthetic dataset to test the effect of label noise. The input x is a 20-dimensional vector where each coordinate is uniformly distributed on $[0, 1]$. The label y is $+1$ if $\sum_{j=1}^{5} x_j \geq 2.5$ and -1 otherwise Mease and Wyner (2008). It also a natural example to validate the ability of an algorithm to learn a sparse feature space.

In our experiment, we use 800 samples as training, 200 as validating, and 1000 as testing. we use the identity feature map $\Phi(\mathbf{x}) = \mathbf{x}$ in all our experiments, and set $t \in \{1.3, 1.6, 1.9\}$ for t-logistic regression. Our main comparator of the t-logistic regression is the logistic regression. The parameter λ is selected in $\{0.1, 0.01, \ldots, 10^{-8}\}$. Label noise is added by randomly choosing 10% or 20% of the labels in the training set and flipping them; each dataset is tested with and without label noise. The convergence criterion is to stop when the change in the objective function value is less than 10^{-8} or a maximum of 3000 function evaluations have been achieved.

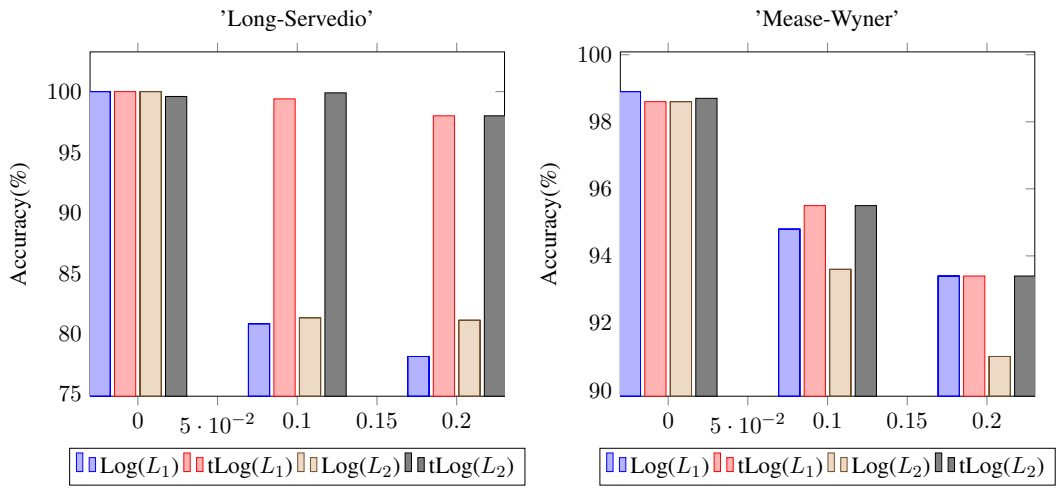


Figure 5: Test results on Long-Servedio and Mease-Wyner datasets.

We plot the test accuracy in Figure 6. It appears to us that the label noise in the Long-Servedio datasets fool the logistic regression, while unable to fool the t-logistic regression. In Mease-Wyner dataset, we notice that in the sparse feature space, t-logistic regression with L_2 regularizer partially benefits from its Student's t-prior which outperforms the L_2-logistic regression with Gaussian prior. While the two algorithms are having closer performance with L_1 regularizer.

To obtain Figure 7 we used the datasets with 10% of label noise, chose the optimal parameter λ in the previous experiment, and plotted the distribution of the $1/z \propto \xi$ obtained after training with $t = 1.9$ and L_2 regularizer. To distinguish the points with noisy labels we plot them in cyan while the other points are plotted in red. Recall that ξ denotes the influence of a point. One can clearly observe that the ξ of the noisy data is much smaller than that of the clean data, which indicates that the algorithm is able to effectively identify these points and cap their influence. In particular, on the Long-Servedio dataset observe the 4 distinct spikes. From left to right, the first spike corresponds to the noisy large margin examples, the second spike represents the noisy pullers, the third spike denotes the clean pullers, while the rightmost spike corresponds to the clean large margin examples. Clearly, the noisy large margin examples and the noisy pullers are assigned a low value of ξ thus capping their influence and leading to the perfect classification of the test set. On the other hand, logistic regression is unable to discriminate between clean and noisy training samples which leads to bad performance on noisy datasets.

9.2 Binary Classification

Datasets Table 2 summarizes the datasets used in our experiments. `adult9`, `astro-ph`, `news20`, `real-sim`, `reuters-c11`, `reuters-ccat` are from the same source as in Hsieh et al. (2008). `aut-avn` is from Andrew McCallum's home page[6], `covertype` is from the UCI repository (Merz and Murphy, 1998), `worm` is from Franc and Sonnenburg (2008), `kdd99` is from KDD

6. http://www.cs.umass.edu/~mccallum/data/sraa.tar.gz.

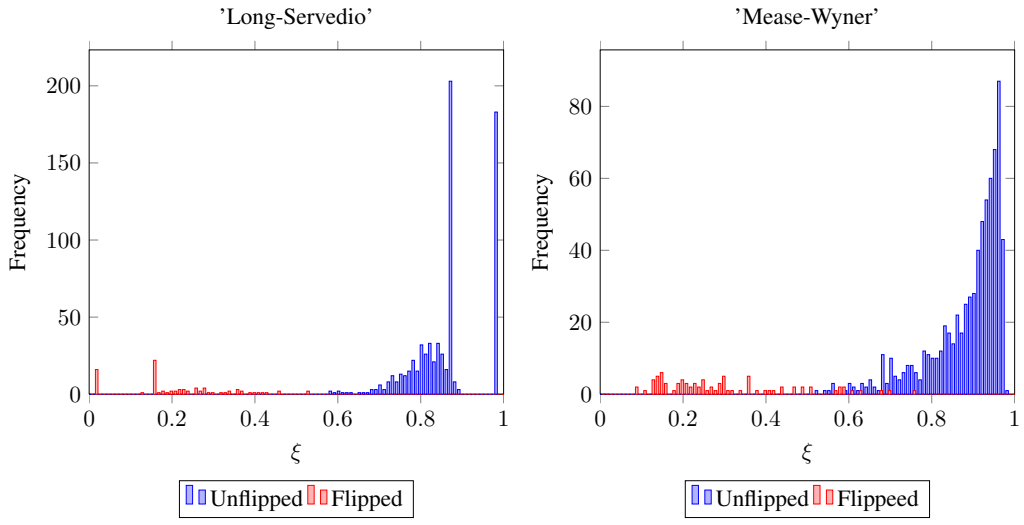


Figure 6: The distribution of ξ obtained after training t-logistic regression with $t = 1.9$. Left: Long-Servedio; Right: Mease-Wyner.

Cup 1999[7], while `web8`, `webspam-u`, `webspam-t`[8], as well as the `kdda` and `kddb`[9] are from the LibSVM binary data collection[10]. The `alpha`, `delta`, `fd`, `gamma`, and `zeta` datasets were all obtained from the Pascal Large Scale Learning Workshop website (Sonnenburg et al., 2008). For the datasets which were also used by Teo et al. (2010) (indicated by an asterisk in Table 2) we used the training test split provided by them, and for the remaining datasets we used 80% of the labeled data for training and the remaining 20% for testing. In order to learn the parameter λ and t, we further partition the training set obtained above into two parts, in which 80% is used for training and the remaining 20% is used for validation. In all cases, we added a constant feature as a bias.

Results We use the same experimental methodology as in the synthetic dataset, and we list the test accuracy results for the four comparators: L_1/L_2 logistic regression and L_1/L_2 t-logistic regression in the table. We also plot the ξ graph of all these datasets, and the black line denotes the ξ value of the decision boundary ($p(y|\,\mathbf{x}, \boldsymbol{\theta}) = 0.5$). Because $\xi \propto p(y|\,\mathbf{x}, \boldsymbol{\theta})^{t-1}$, different t will give different ξ value of the decision boundary. When the points lie on the right side of the boundary, it means that the corresponding $p(y|\,\mathbf{x}, \boldsymbol{\theta}) > 0.5$; on the other hand, if the points are on the left, it means $p(y|\,\mathbf{x}, \boldsymbol{\theta}) < 0.5$.

When label noise is added, the t-logistic cap the influence of the data with label noise (red), and in most of these case, the test accuracy is significantly better than the logistic regression. Notable examples are 'astro-ph', 'aut-avn', 'fd', 'real-sim', 'worm'. For those cases that the test accuracy of

7. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
8. `webspam-u` is the webspam-unigram and `webspam-t` is the webspam-trigram dataset. Original dataset can be found at http://www.cc.gatech.edu/projects/doi/WebbSpamCorpus.html.
9. These datasets were derived from KDD CUP 2010. `kdda` is the first problem algebra_2008_2009 and `kddb` is the second problem bridge_to_algebra_2008_2009.
10. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html.

Table 2: Summary of the datasets used in our experiments. n is the total # of examples, d is the # of features, s is the feature density (% of features that are non-zero), and $n_+ : n_-$ is the ratio of the number of positive vs negative examples. M denotes a million.

dataset	n	d	$s(\%)$	$n_+ : n_-$	**dataset**	n	d	$s(\%)$	$n_+ : n_-$
adult9	48,842	123	11.3	0.32	alpha	500,000	500	100	1.00
astro-ph	94,856	99,757	0.08	0.31	aut-avn	71,066	20,707	0.25	1.84
beta	500,000	500	100	1.00	covertype	581,012	54	22.22	0.57
delta	500,000	500	100	1.00	epsilon	500,000	2000	100	1.00
fd	625,880	900	100	0.09	gamma	500,000	500	100	1.00
kdd99	5.21 M	127	12.86	4.04	kdda	8.92 M	20.22 M	2e−4	5.80
kddb	20.01 M	29.89 M	1e−4	6.18	news20	19,954	7.26 M	0.033	1.00
real-sim	72,201	2.97 M	0.25	0.44	reuters-c11	804,414	1.76 M	0.16	0.03
reuters-ccat	804,414	1.76 M	0.16	0.90	web8	59,245	300	4.24	0.03
webspam-t	350,000	16.61 M	0.022	1.54	webspam-u	350,000	254	33.8	1.54
worm	1.03 M	804	25	0.06	zeta	500,000	800.4 M	100	1.00

t-logistic regression is comparable to the logistic regression, most of time the ξ of data with label noise is also smaller than clean data, which indicates that the flipped data has less influence.

When label noise is not added, the performance of the t-logistic regression and logistic regression seems to be comparable. Additionally, datasets such as 'astro-ph', 'aut-avn', 'fd', 'kdd99', 'news20', 'real-sim', 'reuters-c11', 'web8', 'webspamtrigram', and 'worm', it appears that the ξ value of the training data are almost all concentrated at the area of $p(y|\,\mathbf{x}, \boldsymbol{\theta}) \simeq 1$. It indicates that the weight vector fits nearly perfectly to the training data.

9.3 Multiclass Classification

Datasets Table 5 summarizes the datasets used in our multiclass classification experiments. For the datasets which were also used by Teo et al. (2010) (indicated by an asterisk in Table 2) we used the training test split provided by them, and for the remaining datasets we used 80% of the labeled data for training and the remaining 20% for testing. In order to learn the parameter λ and t, we further partition the training set obtained above into two parts, in which 80% is used for training and the remaining 20% is used for validation. In all cases, we added a constant feature as a bias.

Results We use the same experimental methodology as before, and we list the test errors in the below. To see how t-logistic regression works, we also plot the ξ variables of the results, and the black line denotes the ξ value of the decision boundary ($p(y|\,\mathbf{x}, \boldsymbol{\theta}) = (nc)^{-1}$).

When label noise is added, we observe that almost all the test results of the t-logistic regression are better. And the ξ plots also show that the data with added label noise (red) are distinguished by the algorithm and put far less influence than majority of the clean data (blue).

Without label noise is added, t-logistic regression still works better or as well as logistic regression in most of the dataset. It is observable that when t-logistic regression is better, especially 'letter' and 'mnist' dataset, there are a portion of data get small ξ-values. It is very likely that these are the outliers existed in the origianl dataset, and the capability of removing or capping the influence of these data in t-logistic regression clearly improves the test performance.

Table 3: Test Error on Binary Datasets

Dataset	Log(L_1)	tLog(L_1)	Forget(L_1)	Log(L_2)	tLog(L_2)	Forget(L_2)
adult9(0.0)	14.97	15.03	14.99	14.97	**14.92**	15.11
adult9(0.1)	15.24	**15.16**	15.20	15.28	15.17	15.27
adult9(0.2)	15.47	15.21	**15.16**	15.40	15.28	15.28
alpha(0.0)	21.75	21.77	**21.75**	21.75	21.77	21.80
alpha(0.1)	**21.81**	21.86	21.86	21.84	21.86	21.86
alpha(0.2)	21.87	21.91	21.86	**21.83**	21.85	21.84
astro-ph(0.0)	**0.75**	1.47	1.24	1.02	1.49	1.19
astro-ph(0.1)	3.23	2.37	2.97	2.77	**2.35**	2.62
astro-ph(0.2)	4.77	3.33	3.32	4.22	**3.17**	3.74
aut-avn(0.0)	**1.88**	1.94	1.97	1.92	1.93	1.99
aut-avn(0.1)	4.03	2.71	2.90	3.68	**2.45**	2.68
aut-avn(0.2)	6.77	4.02	**3.56**	4.29	3.59	3.97
beta(0.0)	49.88	49.99	**49.84**	49.99	49.91	49.99
beta(0.1)	49.95	49.85	49.97	49.90	49.94	**49.85**
beta(0.2)	50.01	**49.93**	49.98	50.11	49.95	50.11
covertype(0.0)	23.00	22.63	**22.47**	23.03	22.62	22.49
covertype(0.1)	23.20	22.81	22.56	23.15	22.74	**22.50**
covertype(0.2)	23.32	23.08	22.76	23.24	22.95	**22.59**
delta(0.0)	21.54	21.55	21.54	**21.54**	21.54	21.54
delta(0.1)	**21.51**	21.56	21.55	21.52	21.55	21.53
delta(0.2)	21.68	21.66	**21.65**	21.68	21.67	21.66
epsilon(0.0)	10.21	10.24	**10.19**	10.23	10.23	10.23
epsilon(0.1)	**10.23**	10.33	10.25	10.45	10.38	10.33
epsilon(0.2)	10.59	10.53	**10.50**	10.85	10.70	10.67
fd(0.0)	2.96	3.00	2.92	2.94	**2.86**	2.86
fd(0.1)	3.85	3.22	3.07	3.84	2.99	**2.88**
fd(0.2)	4.30	3.59	3.18	4.29	3.32	**2.98**
gamma(0.0)	19.99	**19.98**	20.00	20.01	19.98	19.98
gamma(0.1)	20.12	20.09	**20.08**	20.11	20.08	20.11
gamma(0.2)	20.14	20.14	20.11	20.21	**20.10**	20.12
kdd99(0.0)	**8.00**	8.15	8.11	8.00	8.28	8.10
kdd99(0.1)	8.16	8.16	8.22	**8.04**	8.12	8.20
kdd99(0.2)	**8.05**	8.19	8.11	8.07	8.07	8.09
kdda(0.0)	10.50	10.58	10.65	11.56	10.50	**10.49**
kdda(0.1)	11.32	10.70	**10.56**	10.73	10.61	10.80
kdda(0.2)	10.70	10.73	**10.67**	10.81	10.77	10.80
kddb(0.0)	10.97	**10.08**	10.08	10.33	10.16	10.28
kddb(0.1)	10.38	10.31	10.32	10.25	10.34	**10.22**
kddb(0.2)	11.01	10.55	10.53	10.61	**10.49**	10.59

Table 4: Test Error on Binary Datasets (Continue)

Dataset	Log(L_1)	tLog(L_1)	Forget(L_1)	Log(L_2)	tLog(L_2)	Forget(L_2)
news20(0.0)	**3.25**	3.48	3.25	3.78	3.88	3.58
news20(0.1)	7.41	**5.53**	6.18	5.98	5.58	6.03
news20(0.2)	14.45	9.16	12.34	7.84	**7.71**	7.84
real-sim(0.0)	2.85	2.70	**2.69**	2.83	2.85	2.83
real-sim(0.1)	4.85	3.57	4.23	4.23	**3.35**	3.46
real-sim(0.2)	7.60	4.63	5.08	5.44	**4.31**	4.97
reuters-c11(0.0)	**2.83**	2.89	2.95	2.95	2.84	2.86
reuters-c11(0.1)	2.86	2.94	2.89	3.03	**2.84**	2.93
reuters-c11(0.2)	2.88	**2.84**	2.89	3.00	2.92	3.00
reuters-ccat(0.0)	7.95	**7.18**	7.33	7.68	7.41	7.56
reuters-ccat(0.1)	8.78	7.77	8.20	7.64	**7.53**	7.64
reuters-ccat(0.2)	9.52	8.50	9.53	8.30	7.95	**7.93**
web8(0.0)	1.12	1.05	**0.87**	1.11	1.01	0.99
web8(0.1)	1.29	1.15	**1.05**	1.29	1.21	1.12
web8(0.2)	1.30	1.23	1.23	1.30	**1.21**	1.23
webspamtrigram(0.0)	0.61	0.69	**0.53**	0.54	0.66	0.54
webspamtrigram(0.1)	1.06	0.76	0.70	0.97	0.81	**0.66**
webspamtrigram(0.2)	1.45	1.17	**1.07**	1.41	1.21	1.22
webspamunigram(0.0)	7.20	6.70	**6.53**	7.21	6.83	6.54
webspamunigram(0.1)	7.49	6.88	**6.52**	7.48	6.91	6.56
webspamunigram(0.2)	7.74	7.16	6.70	7.74	7.16	**6.69**
worm(0.0)	1.51	**1.49**	1.50	1.51	1.50	1.50
worm(0.1)	2.63	**1.55**	1.56	2.63	1.58	1.57
worm(0.2)	3.68	1.63	**1.63**	3.68	1.66	1.64
zeta(0.0)	5.69	5.82	**5.61**	11.76	13.33	11.61
zeta(0.1)	6.13	5.97	**5.73**	11.22	13.20	11.28
zeta(0.2)	6.61	6.15	**5.94**	10.94	12.78	11.18

Table 5: Summary of the datasets used in our experiments. n is the total # of examples, d is the # of features, nc is the # of classes, and s is the feature density (% of features that are non-zero). M denotes a million.

dataset	n	d	nc	s(%)	dataset	n	d	nc	s(%)
dna	2,586	182	3	100	letter	15,500	18	26	100
mnist	70,000	782	10	100	protein	21,516	359	3	100
rcv1	534,130	47,238	52	22.22	sensitacoustic	98,528	52	3	100
sensitcombined	98,528	102	3	100	sensitseismic	98,528	52	3	100
usps	9298	258	10	100					

10. Discussion

10.1 Another Way to Optimize the t-logistic Regression

In the discussion of the optimal condition, we obtain the MLE by minimizing $-\log p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta})$. For t-logistic regression, it is also convenient to minimize $p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta})^{1-t}$,

$$P(\boldsymbol{\theta}) \triangleq p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta})^{1-t} = \prod_{i=1}^{m} p(y_i | \, \mathbf{x}_i; \boldsymbol{\theta})^{1-t} \tag{60}$$

$$= \prod_{i=1}^{m} \underbrace{\left(1 + (1-t)(\boldsymbol{\theta}_{y_i}^T \mathbf{x}_i - g_t(\boldsymbol{\theta} \mid \mathbf{x}_i))\right)}_{l_i(\boldsymbol{\theta})} \tag{61}$$

Since $t > 1$, and $g_t(\boldsymbol{\theta} \mid \mathbf{x}_i)$ is convex, it is easy to see that each component $l_i(\boldsymbol{\theta})$ is positive and convex. Therefore, $P(\boldsymbol{\theta})$ becomes the product of a series of positive convex functions $l_i(\boldsymbol{\theta})$.

The optimal solutions to the problem (61) can be obtained by solving the following parametric problem (see Theorem 2.1 of Kuno et al. (1993)):

$$\min_{\boldsymbol{\xi}} \min_{\boldsymbol{\theta}} \quad \mathrm{MP}(\boldsymbol{\theta}, \boldsymbol{\xi}) \triangleq \sum_{i=1}^{m} \xi_i l_i(\boldsymbol{\theta}) \;\; \text{s.t.} \;\; \boldsymbol{\xi} > \mathbf{0}, \;\; \prod_{i=1}^{m} \xi_i \geq 1. \tag{62}$$

$\boldsymbol{\xi}$-Step: Assume that $\boldsymbol{\theta}$ is fixed, and denote $\tilde{l}_i = l_i(\boldsymbol{\theta})$ to rewrite (62) as:

$$\min_{\boldsymbol{\xi}} \quad \mathrm{MP}(\boldsymbol{\theta}, \boldsymbol{\xi}) = \min_{\boldsymbol{\xi}} \quad \sum_{i=1}^{m} \xi_i \tilde{l}_i \quad \text{s.t.} \quad \boldsymbol{\xi} > \mathbf{0}, \;\; \prod_{i=1}^{m} \xi_i \geq 1. \tag{63}$$

Since the objective function is linear in $\boldsymbol{\xi}$ and the feasible region is a convex set, (63) is a convex optimization problem. By introducing a non-negative Lagrange multiplier $\gamma \geq 0$, the partial Lagrangian and its gradient with respect to $\xi_{i'}$ can be written as

$$L(\boldsymbol{\xi}, \gamma) = \sum_{i=1}^{m} \xi_i \tilde{l}_i + \gamma \cdot \left(1 - \prod_{i=1}^{m} \xi_i\right) \tag{64}$$

$$\frac{\partial}{\partial \xi_{i'}} L(\xi, \gamma) = \tilde{l}_{i'} - \gamma \prod_{i \neq i'} \xi_i. \tag{65}$$

Table 6: Test Error on Multiclass Datasets

Dataset	Log(L_1)	tLog(L_1)	Forget(L_1)	Log(L_2)	tLog(L_2)	Forget(L_2)
dna(0.0)	5.36 ± 0.45	**4.91 ± 0.89**	5.85 ± 1.00	5.96 ± 1.08	6.07 ± 0.46	6.68 ± 1.03
dna(0.1)	6.79 ± 0.86	**5.13 ± 0.69**	6.24 ± 1.18	8.12 ± 1.13	6.90 ± 0.65	8.17 ± 1.13
dna(0.2)	8.01 ± 0.81	**5.74 ± 1.28**	6.79 ± 1.57	8.06 ± 0.88	6.74 ± 0.67	8.12 ± 0.93
letter(0.0)	23.07 ± 0.77	19.84 ± 1.33	27.39 ± 0.96	23.04 ± 0.79	**19.78 ± 1.48**	28.80 ± 1.24
letter(0.1)	24.97 ± 0.60	**20.11 ± 1.06**	27.73 ± 1.21	24.94 ± 0.58	20.11 ± 1.13	28.58 ± 1.49
letter(0.2)	26.71 ± 0.90	20.36 ± 1.36	28.05 ± 1.06	26.65 ± 0.88	**20.29 ± 1.21**	28.79 ± 1.00
mnist(0.0)	8.05 ± 0.30	**7.61 ± 0.24**	10.66 ± 0.16	8.00 ± 0.27	7.83 ± 0.26	11.75 ± 0.19
mnist(0.1)	9.41 ± 0.13	**7.79 ± 0.13**	10.78 ± 0.43	9.40 ± 0.21	7.97 ± 0.19	11.87 ± 0.26
mnist(0.2)	10.24 ± 0.25	**7.73 ± 0.27**	11.13 ± 0.41	10.23 ± 0.29	8.13 ± 0.27	12.10 ± 0.22
protein(0.0)	31.69 ± 0.90	31.71 ± 0.70	31.70 ± 0.80	31.64 ± 0.70	**31.56 ± 0.82**	31.59 ± 0.77
protein(0.1)	32.16 ± 0.77	**31.89 ± 0.82**	32.05 ± 0.82	32.02 ± 0.91	31.97 ± 0.97	32.24 ± 0.95
protein(0.2)	32.87 ± 0.99	32.08 ± 0.82	32.36 ± 1.02	32.32 ± 1.02	**31.99 ± 0.87**	32.51 ± 0.77
rcv1(0.0)	7.44 ± 0.10	7.54 ± 0.10	10.80 ± 0.17	**7.12 ± 0.13**	7.43 ± 0.11	32.88 ± 1.43
rcv1(0.1)	8.70 ± 0.06	**7.46 ± 0.09**	10.42 ± 0.43	7.87 ± 0.09	7.59 ± 0.12	32.65 ± 1.41
rcv1(0.2)	9.41 ± 0.14	**7.59 ± 0.12**	10.77 ± 0.51	8.43 ± 0.13	7.61 ± 0.13	30.07 ± 0.33
sensitacoustic(0.0)	31.68 ± 0.14	**28.46 ± 0.14**	31.72 ± 0.19	31.68 ± 0.15	28.76 ± 0.13	32.52 ± 0.17
sensitacoustic(0.1)	31.91 ± 0.17	**28.85 ± 0.11**	31.90 ± 0.15	31.91 ± 0.16	28.97 ± 0.08	32.32 ± 0.17
sensitacoustic(0.2)	32.12 ± 0.14	**29.29 ± 0.05**	32.18 ± 0.18	32.14 ± 0.13	29.33 ± 0.15	32.36 ± 0.28
sensitcombined(0.0)	19.69 ± 0.28	**18.68 ± 0.19**	20.40 ± 0.26	19.70 ± 0.31	18.81 ± 0.24	21.98 ± 0.40
sensitcombined(0.1)	20.37 ± 0.30	**18.74 ± 0.25**	20.60 ± 0.31	20.37 ± 0.29	18.83 ± 0.26	21.58 ± 0.32
sensitcombined(0.2)	20.67 ± 0.40	**18.92 ± 0.24**	20.76 ± 0.50	20.69 ± 0.40	19.05 ± 0.27	21.16 ± 0.35
sensitseismic(0.0)	28.54 ± 0.48	**26.70 ± 0.44**	29.00 ± 0.55	28.52 ± 0.48	26.92 ± 0.35	29.28 ± 0.46
sensitseismic(0.1)	30.15 ± 0.46	**27.03 ± 0.39**	30.05 ± 0.49	30.12 ± 0.48	27.23 ± 0.33	30.21 ± 0.48
sensitseismic(0.2)	30.65 ± 0.53	**27.35 ± 0.40**	30.65 ± 0.57	30.66 ± 0.52	27.66 ± 0.36	30.64 ± 0.53
usps(0.0)	5.95 ± 0.40	5.78 ± 0.48	7.62 ± 0.76	**5.52 ± 0.32**	5.75 ± 0.53	8.19 ± 0.63
usps(0.1)	7.13 ± 0.58	6.51 ± 0.74	7.45 ± 0.49	6.79 ± 0.53	**5.92 ± 0.52**	7.99 ± 0.52
usps(0.2)	7.45 ± 0.34	6.28 ± 0.50	7.60 ± 0.47	7.44 ± 0.32	**5.90 ± 0.35**	8.53 ± 0.58

Setting the gradient to 0 obtains $\gamma = \frac{\tilde{l}_{i'}}{\prod_{i \neq i'} \xi_i}$. Since $\tilde{l}_{i'} > 0$, it follows that γ cannot be 0. By the K.K.T. conditions (Boyd and Vandenberghe, 2004), we can conclude that $\prod_{i=1}^m \xi_i = 1$. This in turn implies that $\gamma = \tilde{l}_{i'} \xi_{i'}$ or

$$(\xi_1, \ldots, \xi_m) = (\gamma/\tilde{l}_1, \ldots, \gamma/\tilde{l}_m), \text{ with } \gamma = \prod_{i=1}^m \tilde{l}_i^{\frac{1}{m}}. \tag{66}$$

where $\xi_i \propto 1/\tilde{l}_i = p(y_i | \mathbf{x}_i, \boldsymbol{\theta})^{t-1}$.

$\boldsymbol{\theta}$-Step: In this step we fix $\boldsymbol{\xi} > 0$ and solve for the optimal $\boldsymbol{\theta}$. This step is essentially the same as logistic regression, except that each component has a weight ξ_i here.

$$\min_{\boldsymbol{\theta}} \ \text{MP}(\boldsymbol{\theta}, \boldsymbol{\xi}) = \min_{\boldsymbol{\theta}} \ \sum_{i=1}^m \xi_i l_i(\boldsymbol{\theta}) \tag{67}$$

and the gradient will be

$$\frac{\partial}{\partial \boldsymbol{\theta}} \text{MP}(\boldsymbol{\theta}, \boldsymbol{\xi}) = (1-t) \sum_{i=1}^m \xi_i (y_i - \mathbb{E}_{q_i}[y]) \, \mathbf{x}_i^T$$

where $q_i \propto p_i^t$, and $p_i = p(y_i | \mathbf{x}_i, \boldsymbol{\theta})$.

10.2 Multinomality

One of the key disadvantage of the non-convex losses is that it could potentially introduce multiple local minima and therefore it creates difficulty to find the global optimum. However, when the non-convex losses are quasi-convex, it is not immediately clear whether the empirical risk of the dataset $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, which is the sum of the losses of these samples, are multimodal or not. In the following, we show that for any nonconvex loss functions under some mild conditions, one can always construct a dataset which has multimodal empirical risk. The following theorem is proved when the feature dimension is 1-d. Generalization to multi-dimension is trivial.

Theorem 10.1 ?? *Assume that $(x, y) \in (\mathbb{R}, \pm 1)$, define the margin $u = \theta \cdot x \cdot y$, consider a loss function $L(\theta; x, y) = L(u)$ which is smooth at $u = 0$. If $\frac{dL(u)}{du}|_{u=0} = L'(0) = -z_0 < 0$, and there exist $u_1 < 0$ and $u_2 > 0$ where $L'(u_i) > -z_0$ for $i = 1, 2$, then there exists a set of data points $\mathbf{x} = \{x_1, \ldots, x_{n+1}\}$, such that their empirical risk $\sum_{i=1}^{n+1} L(\theta, x_i, y_i)$ as a function of θ has at least two local minima.*

Proof First, there exists z such that

$$L'(u_1) \geq -z \ \text{ and } \ L'(u_2) \geq -z, \text{ where } \ z < z_0. \tag{68}$$

Because the loss function is smooth, there exists δ such that

$$L'(u) < -(z + z_0)/2 \ \text{ , where } \ u \in (-\delta, \delta). \tag{69}$$

Define

$$U = \max\{-u_1, u_2\}. \tag{70}$$

we construct a set of data points which consist of $x_{1,\dots,n} = 1$, $x_{n+1} = x$, where

$$x = -\frac{U}{\delta} \tag{71}$$

$$n = -\frac{(z+z_0)x}{2z_0} \tag{72}$$

The gradient of the empirical risk is,

$$H(\theta) = \frac{d}{d\theta}\sum_{i=1}^{n} L(\theta x_i) + L(\theta x_{n+1}) \tag{73}$$

$$= nL'(\theta) + xL'(\theta x) \tag{74}$$

$$= n\left(L'(\theta) + \frac{x}{n}L'(\theta x)\right) \tag{75}$$

$$\text{when } \theta = 0, \quad H(\theta) = n\left(L'(0) - \frac{2z_0}{z+z_0}L'(0)\right) = \frac{z-z_0}{z+z_0}nL'(0) > 0 \tag{76}$$

$$\text{when } \theta = \delta u_1/U, \quad H(\theta) = n\left(L'(\delta u_1/U) - \frac{2z_0}{z+z_0}L'(u_1)\right)$$
$$< n\left(-(z+z_0)/2 + \frac{2z_0}{z+z_0}z\right) = -\frac{(z-z_0)^2}{2(z+z_0)} < 0 \tag{77}$$

$$\text{when } \theta = \delta u_2/U, \quad H(\theta) = n\left(L'(\delta u_2/U) - \frac{2z_0}{z+z_0}L'(u_2)\right)$$
$$< n\left(-(z+z_0)/2 + \frac{2z_0}{z+z_0}z\right) = -\frac{(z-z_0)^2}{2(z+z_0)} < 0 \tag{78}$$

where the inequalities in (77) and (78) is due to (68) and (69) and the definition of U.

Therefore, $H(\theta)$ must pass the x-axis twice in $(\delta u_1/U, \delta u_2/U)$. Since $H(\delta u_1/U) < 0$ and $H(0) > 0$, one local minimum lies in $(\delta u_1/U, 0)$. Since $H(\delta u_2/U) < 0$, the other local minimum lies in $(\delta u_2/U, +\infty)$. ■

To the best of our knowledge, all the existed nonconvex losses satisfy the above theorem. We list some but not all nonconvex losses in Table 7.

Name	Loss Function
Sigmoid	$L(u) = \frac{1}{1+\exp(u)}$
Probit	$L(u) = 0.5 - 0.5 \cdot erf(u)$
Savage	$L(u) = \frac{1}{(1+\exp(2u))^2}$
Tangent	$L(u) = (2\arctan(u) - 1)^2$
t-logistic	$L(u) = -\log\exp_t(\frac{1}{2}u - g_t(u))$

Table 7: The non-convex losses that satisfy the conditions of Theorem **??**

An interesting observation is that the multimodality is somhow related to the robustness of the nonconvex losses. To see that, consider the following 1-dim example. Assuming that we have 30

data points (x_i, y_i). Without loss of generality, we assume that all y_i are equal to 1. All x_i are equal to 1, except one outlier which is equal to -200. We draw the empirical risk function as a function of θ for both logistic loss and t-logistic loss,

$$\begin{cases} \text{(logistic)} & \mathrm{R}_{\mathrm{emp}}(\theta) = \sum_i \log(1 + \exp(-\theta x_i)) \\ (t\text{-logistic}) & \mathrm{R}_{\mathrm{emp}}(\theta) = \sum_i -\log(\exp_t(\frac{1}{2}\theta x_i - g_t(\theta|x_i))) \end{cases} \tag{79}$$

in Figure 8. When no outlier is added, both logistic loss and t-logistic loss has the MLE θ^* which is larger than zero. However, once the outlier is added, We can see that the minimum of logistic regression has a negative sign of θ^* which indicates that optimal condition is severely impacted by the outlier. On the other hand, the t-logistic regression, although exists a similar local minimum to the logistic regression, has the global minimum which is positive and behaves similar to the θ^* without outliers.

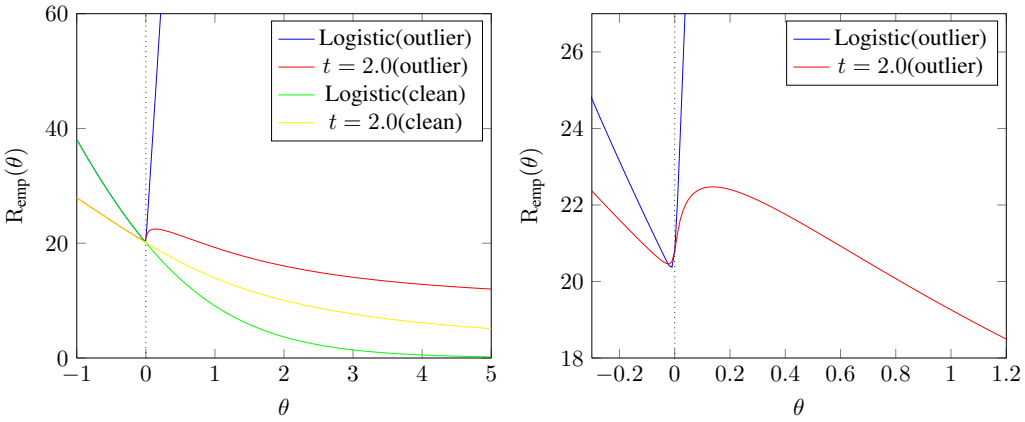


Figure 7: Empirical risk of logistic regression and t-logistic regression

11. Conclusion and Outlook

In this paper, we generalize the logistic regression to the t-logistic regression by using the t-exponential family. The new algorithm has a probabilistic interpretation and is more robust to label noise. We investigate the algorithm in binary classification and multiclass classification, with both L_1 and L_2 regularizer. Although the algorithm is nonconvex, we prove that our algorithm will always converge to a stable point, though it may not always yield to the global optimum. We also prove the t-logistic regression for binary classification is Bayes consistent, and we give a proof showing that nearly all nonconvex losses function could yield to multimodal empirical risks in general.

References

Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, England, 2004.

Vojtěch Franc and Sören Sonnenburg. Optimized cutting plane algorithm for support vector machines. In Andrew McCallum and Sam Roweis, editors, *ICML*, pages 320–327. Omnipress, 2008.

Yoav Freund. A more robust boosting algorithm. Technical Report Arxiv/0905.2138, Arxiv, May 2009.

Cho Jui Hsieh, Kai Wei Chang, Chih Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In William Cohen, Andrew McCallum, and Sam Roweis, editors, *ICML*, pages 408–415. ACM, 2008.

Takahito Kuno, Yasutoshi Yajima, and Hiroshi Konno. An outer approximation method for minimizing the product of several convex functions on a convex set. *Journal of Global Optimization*, 3(3):325–335, September 1993.

Chuanhai Liu. Robit regression: A simple robust alternative to logistic and probit regression. *Applied Bayesian Modeling and Causal Inference from Incomplete-Data Perspectives*, pages 227–238, 2004.

Phil Long and Rocco Servedio. Random classification noise defeats all convex potential boosters. *Machine Learning Journal*, 78(3):287–304, 2010.

David Mease and Abraham Wyner. Evidence contrary to the statistical view of boosting. *Journal of Machine Learning Research*, 9:131–156, February 2008.

C. J. Merz and P. M. Murphy. UCI repository of machine learning databases, 1998. [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

J. Naudts. Deformed exponentials and logarithms in generalized thermostatistics. *Physica A*, 316: 323–334, 2002. URL http://arxiv.org/pdf/cond-mat/0203489.

J. Naudts. Generalized thermostatistics based on deformed exponential and logarithmic functions. *Physica A*, 340:32–40, 2004a.

J. Naudts. Generalized thermostatistics and mean-field theory. *Physica A*, 332:279–300, 2004b.

J. Naudts. Estimators, escort proabilities, and ϕ-exponential families in statistical physics. *Journal of Inequalities in Pure and Applied Mathematics*, 5(4), 2004c.

A O'hagan. On outlier rejection phenomena in bayes inference. *Royal Statistical Society*, 41(3): 358–367, 1979.

Timothy D. Sears. *Generalized Maximum Entropy, Convexity, and Machine Learning*. PhD thesis, Australian National University, 2008.

Soeren Sonnenburg, Vojtech Franc, Elad Yom-Tov, and Michele Sebag. Pascal large scale learning challenge. 2008. URL http://largescale.ml.tu-berlin.de/workshop/.

Choon Hui Teo, S. V. N. Vishwanthan, A. J. Smola, and Quoc V. Le. Bundle methods for regularized risk minimization. *Journal of Machine Learning Research*, 11:311–365, January 2010.

C. Tsallis. Possible generalization of boltzmann-gibbs statistics. *J. Stat. Phys.*, 52:479–487, 1988.