Logistic Regression Trained with Different Loss Functions

Discussion

CS6140

1 Notations

We restrict our discussions to the binary case.

$$g(z) = \frac{1}{1+e^{-z}}$$

$$g'(z) = \frac{\partial g(z)}{\partial z} = g(z)(1-g(z))$$

$$h_w(x) = g(wx) = \frac{1}{1+e^{-wx}} = \frac{1}{1+e^{-\sum_d w^d x^d}}$$

$$P(y=1|x;w) = h_w(x)$$
$$P(y=0|x;w) = 1 - h_w(x)$$

2 Maximum Likelihood Estimation

2.1 Goal

Maximize likelihood:

$$\begin{aligned} L(w) &= p(y|X;w) \\ &= \prod_{i=1}^{m} p(y_i|x_i;w) \\ &= \prod_{i=1}^{m} (h_w(x_i))^{y_i}(1-h_w(x_i))^{1-y_i} \end{aligned}$$

Or equivalently, maximize the log likelihood:

$$\begin{aligned} l(w) &= \log L(w) \\ &= \sum_{i=1}^{m} y_i \log h(x_i) + (1 - y_i) \log(1 - h(x_i)) \end{aligned}$$

2.2 Stochastic Gradient Descent Update Rule

$$\begin{aligned} \frac{\partial}{\partial w^j} l(w) &= (y \frac{1}{g(wx_i)} - (1 - y) \frac{1}{1 - g(wx_i)}) \frac{\partial}{\partial w^j} g(wx_i) \\ &= (y \frac{1}{g(wx_i)} - (1 - y) \frac{1}{1 - g(wx_i)}) g(wx_i)(1 - g(wx_i)) \frac{\partial}{\partial w^j} wx_i \\ &= (y(1 - g(wx_i)) - (1 - y) g(wx_i)) x_i^j \\ &= (y - h_w(x_i)) x_i^j \end{aligned}$$

$$w^j := w^j + \lambda (y_i - h_w(x_i))) x_i^j$$

3 Least Squared Error Estimation

3.1 Goal

Minimize sum of squared error:

$$L(w) = \frac{1}{2} \sum_{i=1}^{m} (y_i - h_w(x_i))^2$$

3.2 Stochastic Gradient Descent Update Rule

$$\begin{aligned} \frac{\partial}{\partial w^j} L(w) &= -(y_i - h_w(x_i)) \frac{\partial h_w(x_i)}{\partial w^j} \\ &= -(y_i - h_w(x_i)) h_w(x_i)(1 - h_w(x_i)) x_i^j \end{aligned}$$

$$w^j := w^j + \lambda (y_i - h_w(x_i)) h_w(x_i)(1 - h_w(x_i)) x_i^j$$

4 Comparison

4.1 Update Rule

For maximum likelihood logistic regression:

$$w^j := w^j + \lambda (y_i - h_w(x_i))) x_i^j$$

For least squared error logistic regression:

$$w^j := w^j + \lambda(y_i - h_w(x_i))h_w(x_i)(1 - h_w(x_i))x_i^j$$

Let

$$f_1(h) = (y - h), y \in \{0, 1\}, h \in (0, 1)$$

$$f_2(h) = (y - h)h(1 - h), y \in \{0, 1\}, h \in (0, 1)$$

When $y = 1$, the plots of $f_1(h)$ and $f_2(h)$ are shown in figure 1. $f_1(h)$ is a monotonously decreasing function, the closer the predicted probability is to 1, the smaller the update is. The farther away the predicted probability is from 1, the bigger the update is. $f_2(h)$ is not monotone. When the predicted probability is close to 0 (far away from the true label), there is almost no update. The biggest update is achieved when $h = \frac{1}{3}$. This behavior seems unintuitive.

One tentative explanation for f_2: f_2 is robust to outliers. In the range of $(\frac{1}{3}, 1)$, the farther away the predicted probability is from 1, the bigger the update is. In this range, f_2 is willing to better incorporate the data point into the model. In the range of $(0, \frac{1}{3})$, the algorithm thinks the predicted value is too far away from the true value, and it's very likely that the data point is mislabeled. In this case, we should give up this data point. The farther away the predicted probability is from 1, the more likely the data point is mislabeled, the less effort we should put on it, thus the smaller the update should be. Therefor $\frac{1}{3}$ can be regarded as a "giving up" threshold.

When $y = 0$, the plots of $f_1(h)$ and $f_2(h)$ are shown in figure 2. $f_1(h)$ is again a monotone function. $f_2(h)$ is not monotone. When the predicted probability is close to 1 (far away from the true label), there is almost no update. The biggest(by absolute value) update is achieved when $h = \frac{2}{3}$. This behavior seems unintuitive. A similar argument based on robustness can be made.

4.2 Consistency

Based on the analysis in [**?**], both loss functions can ensure consistency. The classification errors converge to optimal Bayes error as sample size goes to infinity(if we don't restrict the function class).

4.3 Probability Estimation

4.4 Robustness

If we think logistic regression outputs a probability distribution vector, then both methods try to minimize the distance between the true probability distribution vector and the predicted probability distribution vector. The only difference is in how the distance is defined. For maximum likelihood method, the distance measure is KL-divergence. For least squared error method, the distance is Euclidean distance.

If for a data point x, the true probability distribution is $[0, 1]$, i.e., the true label is 1, but the predicted probability distribution is $[0.99, 0.01]$, then

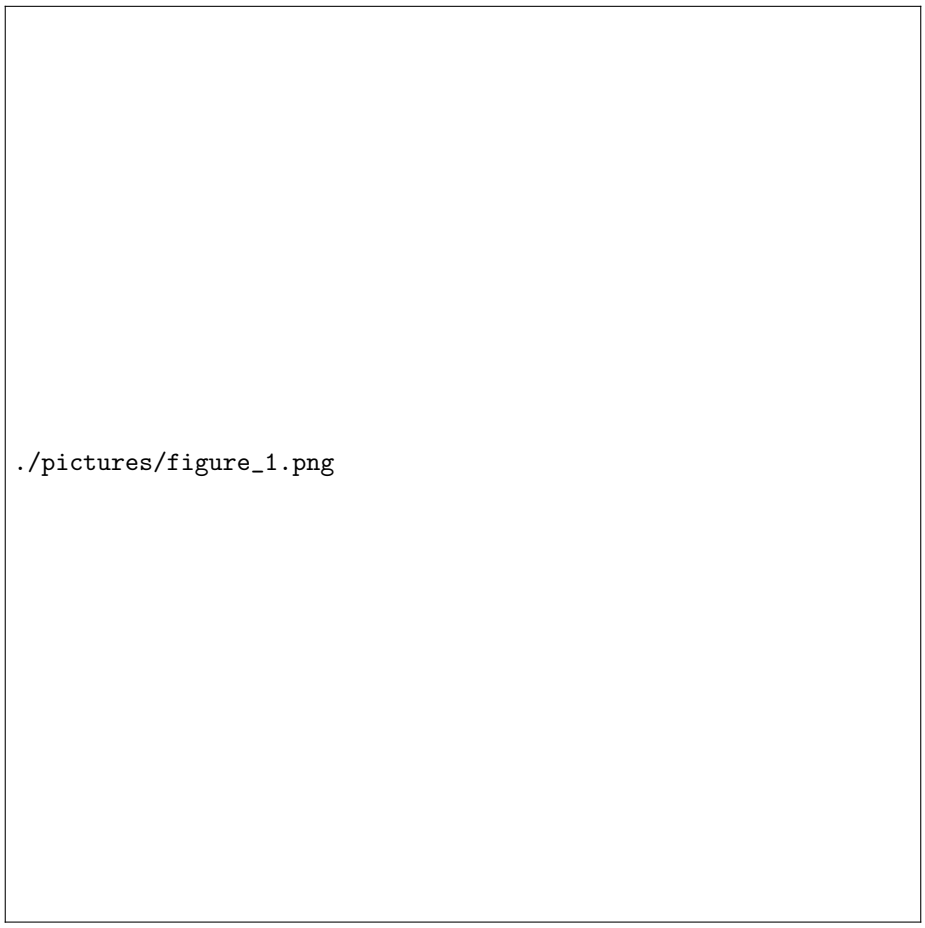

Figure 1: y=1

the KL-divergence between the two distribution is $log_2 100 = 6.64$; the squared Euclidean distance between the two distribution is $0.99^2 + 0.99^2 = 1.96$.

Since KL-divergence is unbounded, but (squared) Euclidean distance has an upper bound 2, it seems logistic regression trained by minimizing squared error is more robust to outliers.

4.5 Convexity and Convergence

The loss function for maximum likelihood estimation is

$$C_1(w) = -l(w) = -\sum_{i=1}^{m} y_i \log h(x_i) + (1 - y_i) \log(1 - h(x_i))$$

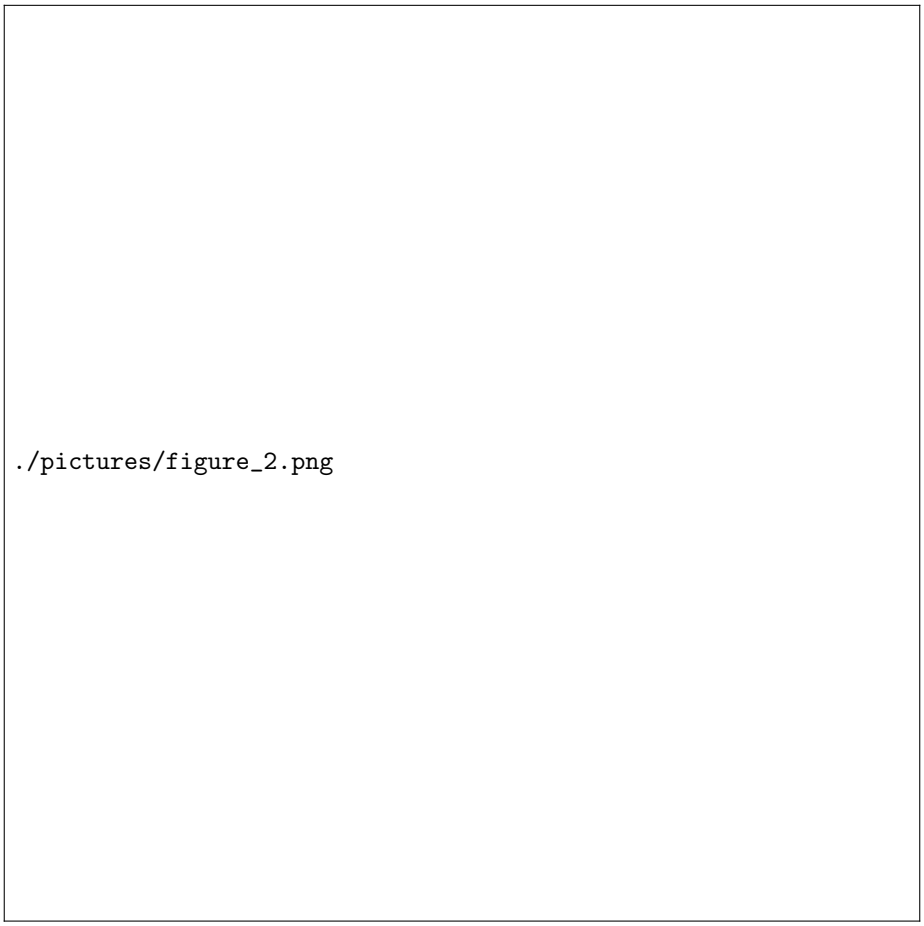

Figure 2: y=0

$C_1(w)$ is a convex function. A proof for the convexity can be found at
http://mathgotchas.blogspot.com/2011/10/why-is-error-function-minimized-in.html

For one data point $x = (1, 1), y = 1$, the plot of $C_1(w)$ is shown in figure 3.

The loss function for the least square estimation is

$$C_2(w) = L(w) = \frac{1}{2}\sum_{i=1}^{m}(y_i - h_w(x_i))^2$$

$C_2(w)$ is a non-convex function. For one data point $x = (1, 1), y = 1$, the plot of $C_2(w)$ is shown in figure 4.

So we can see, training logistic regression by maximizing the likelihood is a convex optimization problem, which is easier; training logistic regression by

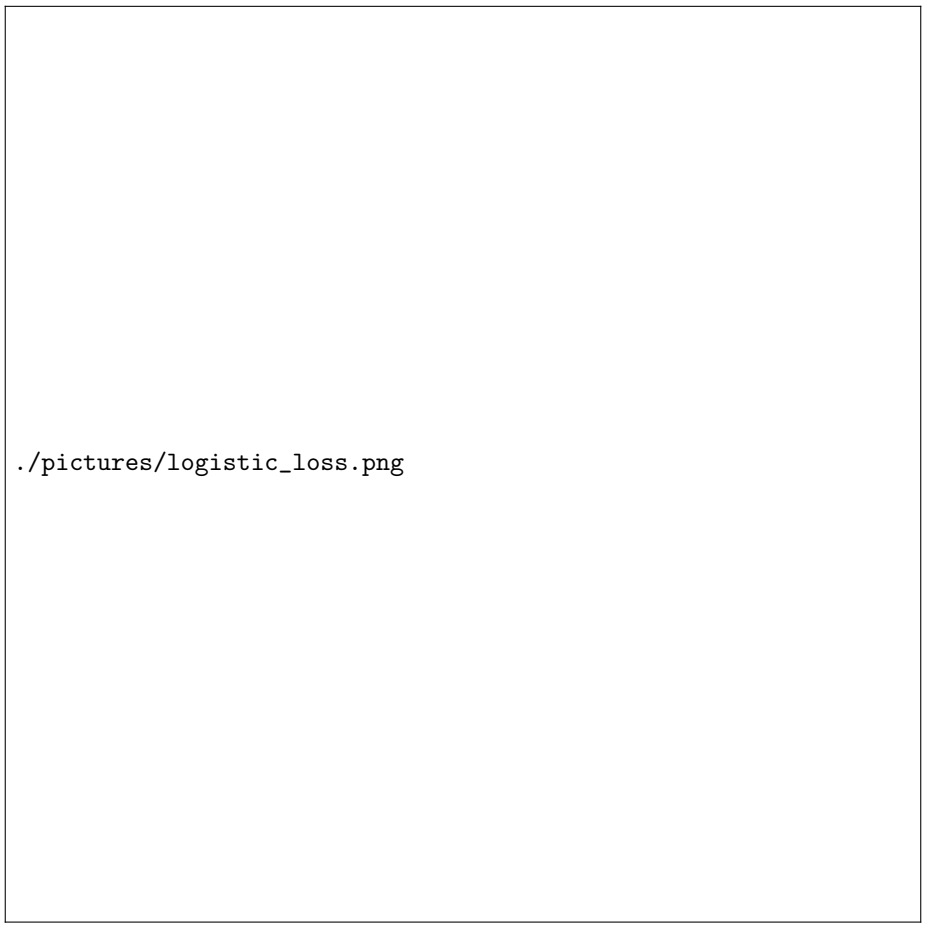

Figure 3: $C_1(w)$

minimizing squared error is a non-convex optimization problem, which is harder.

If we think logistic regression trained with least square error as single node neural network, then it's easy to imagine that the same non-convexity problem would also arise in multi-layer neural networks. Regarding convexity vs non-convexity in machine learning, there is an interesting discussion:
`https://cs.nyu.edu/~yann/talks/lecun-20071207-nonconvex.pdf`

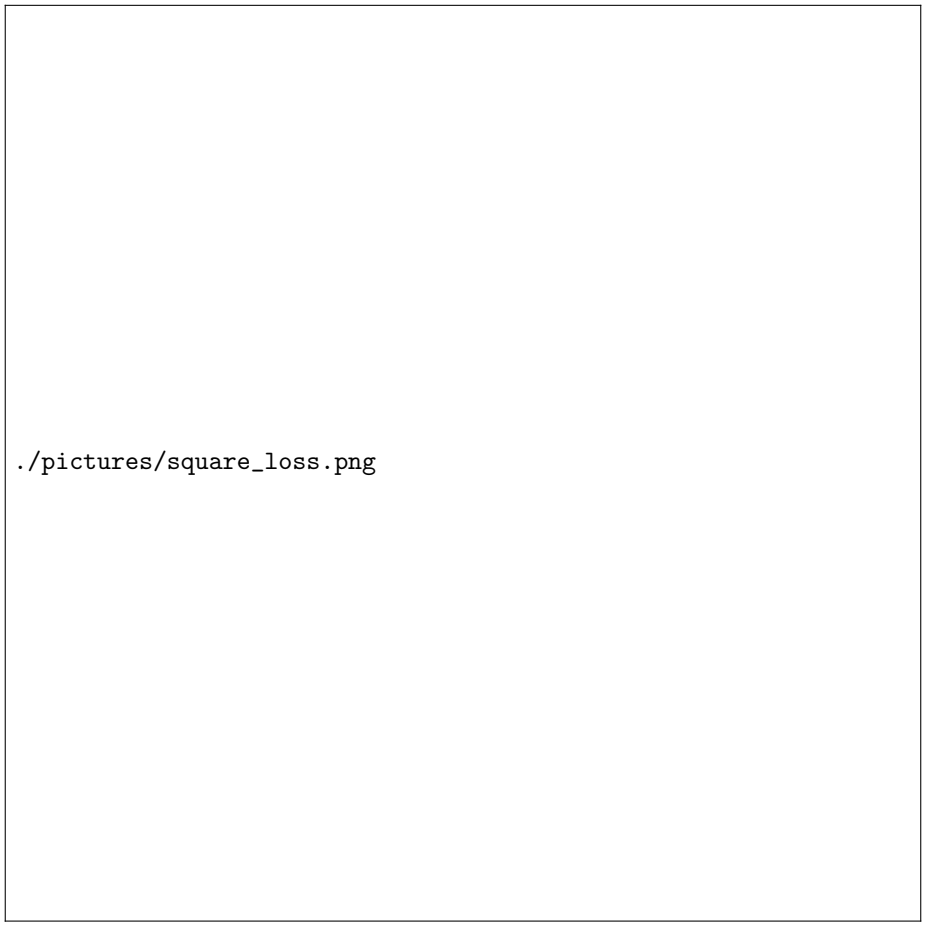

Figure 4: $C_2(w)$

4.6 Newton Method

5 Empirical Study

6 Conclusion