CS5487 Problem Set 8
Linear Classifiers

Antoni Chan
Department of Computer Science
City University of Hong Kong

Logisitic Regression

Problem 8.1 Logistic sigmoid

Let $\sigma(a)$ be the logistic sigmoid function,

$$\sigma(a) = \frac{1}{1+e^{-a}}. \tag{8.1}$$

Let's derive some useful properties:

(a) Show that the derivative of the sigmoid is

$$\frac{\partial \sigma(a)}{\partial a} = \sigma(a)(1-\sigma(a)) \tag{8.2}$$

(b) Show that

$$1 - \sigma(f) = \sigma(-f). \tag{8.3}$$

(c) Show that the inverse of $\sigma(a)$ is

$$\sigma^{-1}(a) = \log \frac{a}{1-a}. \tag{8.4}$$

This is called the *logit* function, or *log odds*.

.....

Problem 8.2 Logistic Regression: MLE and IRLS

Consider the two-class logistic regression problem. Given the training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ is the input and $y_i \in \{0, 1\}$ is the corresponding class. Define the conditional probability of the output class given the input

$$p(y_i|x_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i}, \tag{8.5}$$

where $\pi_i = \sigma(w^T x_i)$ is the conditional probability that x_i belongs to class 1, and $\sigma(a)$ is the logistic sigmoid function,

$$\sigma(a) = \frac{1}{1+e^{-a}}. \tag{8.6}$$

In this problem, we will derive the maximum likelihood estimate of w from the training set $\mathcal{D}$,

$$w^* = \operatorname*{argmax}_w \ell(w), \quad \ell(w) = \sum_{i=1}^{n} \log p(y_i|x_i), \tag{8.7}$$

or equivalently

$$w^* = \operatorname*{argmin}_w E(w), \tag{8.8}$$

$$E(w) = \sum_{i=1}^{n} -\log p(y_i|x_i) = \sum_{i=1}^{n} -\left\{y_i \log \pi_i + (1-y_i)\log(1-\pi_i)\right\}. \tag{8.9}$$

Define $X = [x_1, \cdots, x_n]$, $y = [y_1, \cdots, y_n]^T$, and $\pi = [\pi_1, \cdots, \pi_n]^T$.

(a) Show that the gradient of $E(w)$ is

$$\nabla E(w) = \sum_{i=1}^{n} (\pi_i - y_i) x_i = X(\pi - y). \tag{8.10}$$

Hint: use (8.2).

(b) Show that the Hessian of $E(w)$ is

$$\nabla^2 E(w) = \sum_{i=1}^{n} \pi_i(1-\pi_i) x_i x_i^T = XRX^T, \tag{8.11}$$

where $R = \operatorname{diag}(\pi_1(1-\pi_1), \cdots, \pi_n(1-\pi_n))$.

(c) Show that $\nabla^2 E(w)$ is strictly positive definite, and hence $E(w)$ is convex and has a unique minimum.

(d) $E(w)$ can be minimized using the Newton-Raphson method (see Problem 8.6), which iteratively updates w,

$$w^{(new)} = w^{(old)} - [\nabla^2 E(w)]^{-1} \nabla E(w). \tag{8.12}$$

Show that the update step for logistic regression is

$$w^{(new)} = (XRX^T)^{-1} XRz, \tag{8.13}$$

where R and z are calculated from the previous $w^{(old)}$,

$$R = \operatorname{diag}(\pi_1(1-\pi_1), \cdots, \pi_n(1-\pi_n)), \tag{8.14}$$

$$z = X^T w^{(old)} - R^{-1}(\pi - y). \tag{8.15}$$

The formula in (8.13) is the same as that of weighted least-squares, with weighting R and target z. The weighting and target change in each iteration, since they depend on the current estimate of w. Hence this algorithm is called *iterative reweighted least-squares* (IRLS or IRWLS).

. . . . . .

Problem 8.3 Logistic Regression: Overfitting and singularities

In this problem, we will consider how logistic regression (Problem 8.2) can overfit to a training set when the points are linearly separable.

(a) Show that for a linearly separable training set, the ML solution for logistic regression is obtained by finding a w whose decision boundary $w^T x = 0$ separates the classes, and then taking the magnitude of w to infinity.

(b) In this case, what happens to the shape of the sigmoid function, and the resulting estimates of the posterior probabilities $p(y|x)$? How is this a case of overfitting?

This singularity problem can be fixed by including a prior and finding the MAP solution, as considered in Problem 8.4. The prior term effectively adds a penalty on w with large magnitude, $\|w\|^2$, thus preventing its tendency to infinity.

.....

Problem 8.4 Regularized Logistic Regression: MAP framework

In this problem we will consider the MAP estimation for logistic regression. This is also known as *regularized logistic regression.*

Assume a prior distribution on w that is zero-mean Gaussian and known precision matrix Γ (i.e., inverse of the covariance matrix),

$$p(w) = \mathcal{N}(w|0, \Gamma^{-1}). \tag{8.16}$$

Given the training set $\mathcal{D}$, the MAP estimate is

$$w^* = \underset{w}{\operatorname{argmax}} \log p(y|X, w) + \log p(w). \tag{8.17}$$

(a) Show that (8.17) is equivalent to the minimization problem,

$$w^* = \underset{w}{\operatorname{argmin}} \hat{E}(w), \tag{8.18}$$

$$\hat{E}(w) = E(w) + \frac{1}{2} w^T \Gamma w. \tag{8.19}$$

with $E(w)$ defined in (8.9).

(b) Show that the gradient of $\hat{E}(w)$ is given by

$$\nabla \hat{E}(w) = X(\pi - y) + \Gamma w. \tag{8.20}$$

(c) Show that the Hessian of $\hat{E}(w)$ is given by

$$\nabla^2 \hat{E}(w) = XRX^T + \Gamma, \tag{8.21}$$

with $R = \operatorname{diag}(\pi_1(1-\pi_1), \cdots, \pi_n(1-\pi_n))$ as before.

(d) Show that the Newton-Raphson iteration (see Problem 8.2d and Problem 8.6) for minimizing $\hat{E}(w)$ is

$$w^{(new)} = (XRX^T + \Gamma)^{-1} XRz, \tag{8.22}$$

where R and z are calculated from the previous $w^{(old)}$,

$$R = \text{diag}(\pi_1(1-\pi_1), \cdots, \pi_n(1-\pi_n)), \tag{8.23}$$

$$z = X^T w^{(old)} - R^{-1}(\pi - y). \tag{8.24}$$

(e) How does the precision Γ help to smoothen (regularize) the estimate of w?

(f) Consider the case where we include the bias term in w and x, i.e.

$$x = \begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix}, \quad w = \begin{bmatrix} \tilde{w} \\ \tilde{b} \end{bmatrix}, \tag{8.25}$$

where $\tilde{x}$ is the original input feature vector. The linear discriminant function now contains a bias term

$$f(x) = w^T x = \tilde{w}^T \tilde{x} + \tilde{b}. \tag{8.26}$$

For the regularization, consider the two precision matrices

$$\Gamma_1 = \lambda I, \tag{8.27}$$

$$\Gamma_2 = \text{diag}(\lambda, \cdots, \lambda, 0). \tag{8.28}$$

Show that Γ_1 applies a penalty based on the magnitudes of $\tilde{w}$ and $\tilde{b}$, while Γ_2 applies a penalty only on the magnitude of $\tilde{w}$. In general, should we prefer Γ_1 or Γ_2? Why? (Hint: consider translations of the training set in the input space).

.....

Empirical Risk Minimization

Problem 8.5 Loss functions

All the linear classification algorithms that we saw in lecture are optimization problems over some error function on the training set. Where they differ is in how they define this error function.

Consider the 2-class problem, with $y \in \{+1, -1\}$, input $x \in \mathbb{R}^d$, and training set $\{(x_i, y_i)\}_{i=1}^n$. Define the *emprical risk* over the training set as

$$R_{emp}(w) = \sum_{i=1}^{n} L(f(x_i), y_i), \tag{8.29}$$

where $f(x_i) = w^T x_i$ and $L(\cdot)$ is a loss function. The optimal separating hyperplane is obtained by minimizing the empirical risk,

$$w^* = \underset{w}{\text{argmin}}\, R_{emp}(w). \tag{8.30}$$

This is called *empirical risk minimization.*

In this problem, we will consider the classifiers from lecture within this framework. Define the quantity

$$z_i = y_i w^T x_i. \tag{8.31}$$

Recall that $z_i > 0$ when the training point x_i is correctly classified, and $z_i < 0$ when the point is misclassified.

(a) Show that the loss function for minimizing the number of misclassified training points is

$$L_{01}(z_i) = \begin{cases} 0, & z_i \geq 0 \\ 1, & z_i < 0, \end{cases} \tag{8.32}$$

i.e., the 0-1 loss function.

(b) Show that the loss-function for the perceptron is

$$L_p(z_i) = \max(0, -z_i). \tag{8.33}$$

(c) Show that the loss-function for least-squares classification is

$$L_{LSC}(z_i) = (z_i - 1)^2. \tag{8.34}$$

Hint: use the fact that $y_i^2 = 1$.

(d) Show that the loss-function for logistic regression is

$$L_{LR}(z_i) \propto \frac{1}{\log(2)} \log(1 + e^{-z_i}). \tag{8.35}$$

Hint: use (8.3).

(e) Plot the above loss functions as a function of z_i. Intuitively, which loss functions should be better for learning a good linear classifier? Why?

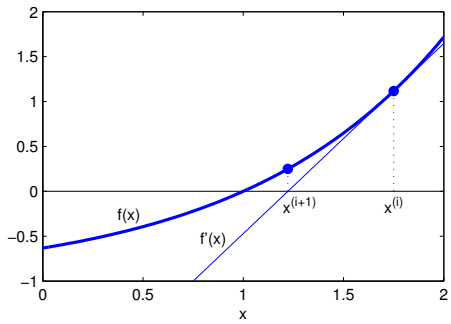
.....

Optimization

Problem 8.6 Newton-Raphson Method

The Newton-Raphson method (also called Newton's method) is an iterative scheme for finding a root (or zero) of a function $f(x)$, i.e., an x^* such that $f(x^*) = 0$.

(a) Given an initial point $x^{(0)}$, the Newton-Raphson method constructs a tangent to $f(x)$ at the current point $x^{(i)}$ and then sets the next $x^{(i+1)}$ to the zero of the tangent function.



Show that this scheme yields the iteration,

$$x^{(i+1)} = x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})}, \tag{8.36}$$

where $f'(x)$ is the first derivative of $f(x)$.

(b) For well-behaved function, the Newton-Raphson method converges quadratically to the zero, if it is started "sufficiently" close to the true x^*. However, the iterations are not guaranteed to converge and could oscillate in an infinite cycle. Consider the function $f(x) = x^3 - 2x + 2$. Plot the function, and show that when starting at $x = 0$, the Newton-Raphson iterations never converge.

(c) The Newton-Raphson method can also be used to find an optimal point in a function. Given a function $g(x)$, show that an optimal point (or stationary point) can be found by the iterations,

$$x^{(i+1)} = x^{(i)} - \frac{g'(x^{(i)})}{g''(x^{(i)})}, \tag{8.37}$$

where $g'(x)$ and $g''(x)$ are the first and second derivatives of $g(x)$. How is the iteration in (8.37) similar to gradient ascent? Does it need to be modified to do gradient descent?

(d) Show that the iteration in (8.37) is equivalent to successively optimizing a quadratic approximation of $g(x)$.

.....