LECTURE 8
Regularized logistics regression

One drawback with the SVM is that the method does not explicitly output a probability or likelihood of the labels, instead the output is a real value the magnitude of which should be monotonic with respect to the condition probability where H is a monotonic function

$$P(y = \pm 1|x) = H(y\, f(x)).$$

This issue can be addressed by using a loss function based upon logistic or binary regression. The main idea behind logistic regression is that we are trying to model the log likelihood ratio by the function $f(x)$

$$f(x) = \log\left(\frac{P(y=1|x)}{P(y=-1|x)}\right).$$

Since $P(y = 1|x)$ is a Bernoulli random variable we can rewrite the above equation as

$$\begin{aligned} f(x) &= \log\left(\frac{P(y=1|x)}{P(y=-1|x)}\right) \\ &= \log\left(\frac{P(y=1|x)}{1-P(y=1|x)}\right) \end{aligned}$$

which implies

$$\begin{aligned} P(y=1|x) &= \frac{1}{1+\exp(-f(x))} \\ P(y=-1|x) &= \frac{1}{1+\exp(f(x))} \\ P(y=\pm 1|x) &= \frac{1}{1+\exp(-y\, f(x))}. \end{aligned}$$

Given a data set $D = \{(x_i, y_i)\}_{i=1}^n$ and a class of functions $f \in \mathcal{H}$ the maximum likelihood estimator (MLE) is the function that maximizes the likelihood of observing the data set D

$$f^*_{\text{MLE}} := \arg\max_{\beta \in \mathbb{R}^p} [P(D|f)] = \arg\max_{\beta \in \mathbb{R}^p} \left[\prod_{i=1}^n \frac{1}{1+\exp(-y_i\, \beta^T x_i)}\right].$$

As in the case of Empirical risk minimization the MLE estimate may overfit the data since there is no smoothness or regularization term. A classical way of imposing smoothness in this context is by placing a prior on β

$$P(\beta) \propto e^{-\kappa \|\beta|^2}.$$

Given a prior and a likelihood we can use Bayes rule to compute the posterior distribution $P(f|D)$

$$P(\beta|D) = \frac{P(D|\beta)\, P(\beta)}{P(D)}.$$

If we plug the prior and likelihood into Bayes rule we can compute the maximum a posteriori (MAP) estimator

$$\begin{aligned}
\beta^*_{\text{MAP}} &:= \arg\max_{\beta \in \mathbb{R}^p} \left[\frac{P(D|\beta)P(\beta)}{P(D)} \right] \\
&= \arg\max_{f \in \mathcal{H}} \left[\frac{\prod_{i=1}^n \frac{1}{1+\exp(-y_i\, \beta^T x_i)}\, e^{-\kappa \|\beta\|^2} P(D)}{} \right] \\
&= \arg\max_{\beta \in \mathbb{R}^p} \left[\sum_{i=1}^n \log\left(\frac{1}{1+\exp(-y_i\, \beta^T x_i)} \right) - \kappa \|\beta\|^2 \right].
\end{aligned}$$

With some simple algebra the above MAP estimator can be rewritten in the form of Tikhonov regularization

$$f^*_{\text{MAP}} = \arg\min_{\beta \in \mathbb{R}^p} \left[n^{-1} \sum_{i=1}^n \log(1+\exp(-y_i\, \beta^T x_i)) + \lambda \|\beta\|^2 \right],$$

where λ is the regularization parameter.

This optimization problem is convex and differentiable so a classical approach for solving for β is using the Newton-Raphson method.

8.1. Newton-Raphson

The Newton-Raphson method was initially used to solve for roots of polynomials and the application to optimization problems is fairly starightforward. We first describe the Newton-Raphson method for the case of a scalar, the optimization is in terms of one variable. We then describe the multivariate form and apply this to the optimization problem in logistic regression.

- Newton's method for finding roots: Newton's method is primarily a method for finding roots of polynomials. It was proposed by Newton around 1669 and Raphson improved on the method in 1690, therefore the Newton-Raphson method. Given a polynomial $f(x)$ the Taylor series expansion of $f(x)$ around the point $x = x_0 + \varepsilon$ is given by

  $$f(x_0 + \varepsilon) = f(x_0) + f'(x_0)\varepsilon + \frac{1}{2} f''(x_0)\varepsilon^2 + ...$$

  truncating the expansion after first order terms results in

  $$f(x_0 + \varepsilon) \approx f(x_0) + f'(x_0)\varepsilon.$$

  From the above expression we can estimate the offset ε needed to get closer to the root ($x :\ f(x) = 0$) starting from the intial guess x_0. This is

done by setting $f(x_0 + \varepsilon) = 0$ and solving for ε.

$$\begin{aligned} 0 &= f(x_0 + \varepsilon) \\ 0 &\approx f(x_0) + f'(x_0)\varepsilon \\ -f(x_0) &\approx f'(x_0)\varepsilon \\ \epsilon_0 &\approx -\frac{f(x_0)}{f'(x_0)}. \end{aligned}$$

This is the first order or linear adjustment to the root's position. This can be turned into an iterative procedure by setting $x_1 = x_0 + \varepsilon_0$, calculating a new ε_1 and then iterating until converegence:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

- The Newton-Raphson method as an optimization method for scalars: We are given a convex minimization problem

$$\min_{x \in [a,b]} g(x),$$

where $g(x)$ is a convex function. The extrema of $g(x)$ will occur at a value of x_m such that $g'(x_m) = 0$ and since the function is convex this extrema will be a minima. If $g(x)$ is a polynomial then $g'(x)$ is also a poynomial and we can apply Newton's method for root finding to $g'(x)$. If $g(x)$ is not a polynomial then we apply the root finding method to a polynomial approximation of $g(x)$. We now describe the steps involved.

(1) Taylor expand $g(x)$: A truncated Taylor expansion of $g(x)$ results in a second order polynomial approximation of $g(x)$

$$g(x) \approx g(x_0) + g'(x_0)(x - x_0) + \frac{1}{2}(x - x_0)^2 g''(x_0).$$

(2) Set derivative to zero: Take the derivative of the Taylor expansion and set it equal to zero

$$\frac{dg}{dx} = f(x) = g'(x_0) + g''(x_0)(x - x_0) = 0.$$

This leaves us with with a root finding problem, find the root of $f(x)$ for which we we use Newton's method for finding roots.

(3) Update rule: The update rule reduces to

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{g'(x_n)}{g''(x_n)}.$$

A key point in the above procedure is the convexity of $g(x)$. To be sure that the procedure converges the second derivative $g''(x)$ must be positive in the domain of optimization, the interval $[a, b]$. Convexity of $g(x)$ ensures this.

- The Newton-Raphson method as an optimization method for vectors: We are given a convex minimization problem

$$\min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}),$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ is convex and $g(\mathbf{x})$ is a convex function. We follow the logic of the scalar case except using vector calculus.

(1) Taylor expand $g(\mathbf{x})$: A truncated Taylor expansion of $g(\mathbf{x})$ results in a second order polynomial approximation of $g(\mathbf{x})$

$$g(\mathbf{x}) \approx g(\mathbf{x}_0) + (\mathbf{x} - \mathbf{x}_0)^T \cdot \nabla g(\mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \cdot \mathbf{H}(x_0) \cdot (\mathbf{x} - \mathbf{x}_0),$$

where $\mathbf{x}$ is a column vector of length n, $\nabla g(\mathbf{x}_0)$ is the gradient of g evaluated at $\mathbf{x}_0$ and is also a column vector of length n, $\mathbf{H}(x_0)$ is the Hessian matrix evaluated at $\mathbf{x}_0$

$$\mathbf{H}_{i,j}(x_0) = \frac{\partial^2 g(\mathbf{x})}{\partial \mathbf{x}^i \partial \mathbf{x}^j}\Big|_{\mathbf{x}_0}, \quad i, j = 1, ..., n.$$

(2) Set derivative to zero: Take the derivative of the Taylor expansion and set it equal to zero

$$\nabla g(\mathbf{x}) = \nabla g(\mathbf{x}_0) + \frac{1}{2}\mathbf{H}(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \cdot \mathbf{H}(\mathbf{x}_0) = 0,$$

the Hessian matrix is symmetric and twice differentiable (due to convexity) so we can reduce the above to

$$\nabla g(\mathbf{x}) = \nabla g(\mathbf{x}_0) + \mathbf{H}(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0) = 0.$$

This implies that at a minima $\mathbf{x}^*$, the gradient is zero

$$0 = \mathbf{H}(\mathbf{x}_0) \cdot (\mathbf{x}^* - \mathbf{x}_0) + \nabla g(\mathbf{x}_0).$$

(3) Update rule: Solving the above linear system of equations for $\mathbf{x}^*$ leads to the following update rule

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{H}^{-1}(\mathbf{x}_n) \cdot \nabla g(\mathbf{x}_n),$$

where $-\mathbf{H}^{-1}(\mathbf{x}_n) \cdot \nabla g(\mathbf{x}_n)$ is called the Newton direction.

For the above procedure to converge to a minima the Newton direction must be a direction of descent

$$\nabla g^T(\mathbf{x}_n) \cdot (\mathbf{x}_{n+1} - \mathbf{x}_n) < 0.$$

If the Hessian matrix is positive definite then the Newton direction will be a direction of descent, this is the matrix analog of a positive second derivative. Convexity of $g(\mathbf{x})$ in the domain $\mathcal{X}$ ensures that the Hessian is positvie definite. If the function $g(\mathbf{x})$ is quadratic the procedure will converge in one iteration.