Statistical Learning 3

RG chapter 4, sections 1, 2, 3 and 4

Jakob Söhl

Delft University of Technology

Binary responses – Classification

Binary response

Assume the measurement is binary: $Y \in \{0, 1\}$.

Suppose data:

- $x_i \in \mathbb{R}^p$ (features);
- $Y_i \in \{0, 1\}$ (responses/measurements).

Aims

1. find a relation between x_i and Y_i;
2. predict Y_{new} for x_{new}.

Logistic regression model

Define $\psi : \mathbb{R} \to (0,1)$ by

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

Assume

$$Y_i \mid \theta \overset{\text{ind}}{\sim} Ber\left(\psi(\theta^T x_i)\right),$$

If $p = \psi(\theta^T x)$, then

$$\log \frac{p}{1-p} = \theta^T x \qquad\qquad \text{log odds.}$$

Taking instead $\psi = \Phi$ (cdf of standard normal distribution) yields Probit regression.

Computing MLE or posterior distribution

Analytic tractability is lost.

$$L(\theta) = \prod_{i=1}^{n} p_i^{y_i}(1 - p_i)^{1-y_i}, \qquad \text{with} \qquad p_i = \psi(\theta^T x_i).$$

Some options:

1. Settle with a point estimate (usually the MAP = Maximum A Posteriori estimate).
2. Approximate the posterior with a tractable class of densities (Laplace approximation, Variational Bayes).
3. Use stochastic sampling methods (MCMC=Markov Chain Monte Carlo).

MAP estimator

Suppose prior π on θ. Compute

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \left(\log L(\theta) + \log \pi(\theta) \right).$$

Common approaches:

1. Gradient descent.
2. Stochastic gradient descent.
3. Variations... Active area of research.

Newton's algorithm for optimisation

Newton's method for root-finding

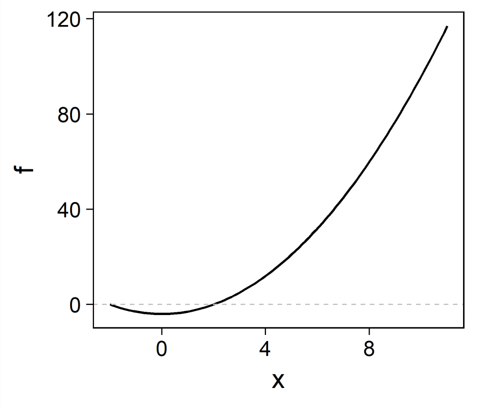

Figure 1: Newton's method for solving $f(x) = x^2 - 4 = 0$.

Formal description

- The task is to find a root θ^* to $f(\theta) = 0$.
- The first-order expansion of f at point θ^j gives

$$0 = f(\theta^*) \approx f(\theta^j) + f'(\theta^j)(\theta^* - \theta^j).$$

- Rearrange to get $\theta^* \approx \theta^j - f(\theta^j)/f'(\theta^j)$.
- This motivates an iterative scheme

$$\theta^{j+1} = \theta^j - \frac{f(\theta^j)}{f'(\theta^j)}.$$

- The convergence is guaranteed if the initial θ^0 is close enough to θ^*. The speed is quadratic:

$$|\theta_{j+1} - \theta^*| \leq C|\theta_j - \theta^*|^2.$$

Newton's method in optimisation

- In the optimisation context, $f = F'$. The iterative scheme is

$$\theta^{j+1} = \theta^j - \frac{F'(\theta^j)}{F''(\theta^j)}.$$

- Newton's method is second-order: it requires the second derivative F'' of F.
- The method can only find a stationary point θ^* of F. For a minimum, check whether

$$F''(\theta^*) > 0.$$

Higher dimensions?

- What happens when the function F depends on n variables: $F(\theta_1, \ldots, \theta_n)$?
- The derivative F' gets replaced with the gradient

$$\nabla F = \begin{pmatrix} \frac{\partial F}{\partial \theta_1} \\ \vdots \\ \frac{\partial F}{\partial \theta_n} \end{pmatrix}.$$

- The second derivative F'' gets replaced with the Hessian matrix H with entries

$$H_{ij} = \frac{\partial^2 F}{\partial \theta_i \partial \theta_j} = H_{ji}, \quad i, j = 1, \ldots, n.$$

$F(x, y) = 0.5x^2 + y^2$

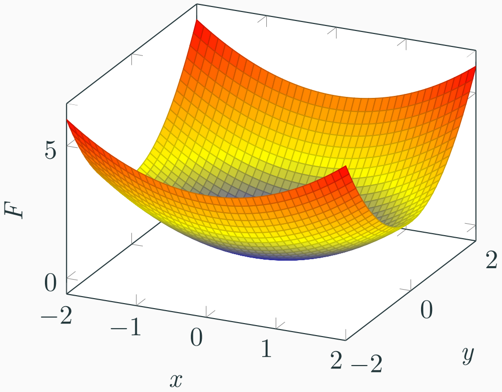

Figure 2: $F(x, y) = 0.5x^2 + y^2$.

Quick computation

- The gradient of $F(x, y) = 0.5x^2 + y^2$ is

$$\nabla F = \begin{pmatrix} x \\ 2y \end{pmatrix}.$$

- The Hessian is

$$H = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}.$$

$F(x, y) = 0.5x^2 + y^2$ and its gradient field

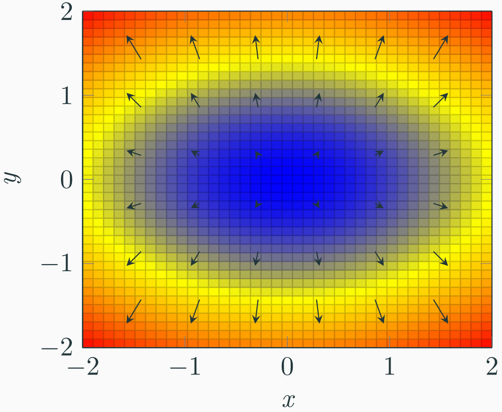

Figure 3: $F(x, y) = 0.5x^2 + y^2$ and its gradient field.

Newton's method: general case

- In vector notation, Newton's method takes the form

$$\theta^{j+1} = \theta^j - (H(\theta^j))^{-1} \nabla F(\theta^j).$$

- The method aims at solving $\nabla F = 0$. A point of minimum is a stationary point.
- The convergence to a stationary point θ^* is fast (quadratic), because the method uses the second derivatives $H_{ij} = \partial^2 F / \partial\theta_i \partial\theta_j$. A minimum is guaranteed, if the matrix $H(\theta^*)$ is positive definite.
- H contains $n(n+1)/2$ distinct elements. Already for $n = 100$ this gives 5050 quantities. Computing 2nd derivatives can be expensive.

Gradient descent

Gradient descent

- Gradient descent is an iterative scheme

$$\theta^{j+1} = \theta^j - s_j \nabla F(\theta^j)$$

  to minimise a function $F(\theta_1, \ldots, \theta_n)$ of $\theta = (\theta_1, \ldots, \theta_n)$.
- The tuning constant $s_j > 0$ is the stepsize. In computer science it is called the learning rate.
- Gradient descent was devised by Cauchy in 1847 to solve nonlinear equations in astronomy.
- The name comes from the fact that in each iteration the method proceeds in the direction opposite to the gradient, i.e., in the direction $-\nabla F$. The Hessian is not required.

Gradient descent: illustration

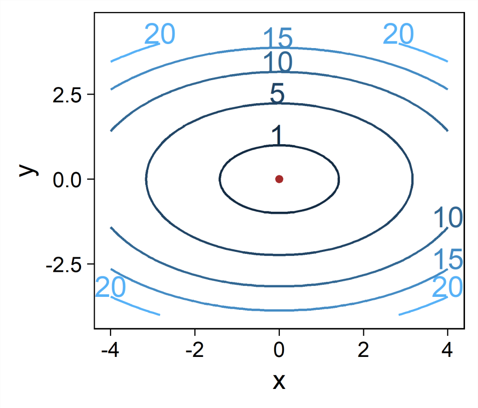

Figure 4: Gradient descent for $F(x) = 0.5x^2 + y^2$. The learning rate equals 0.2.

Moving in the direction of $-F'$

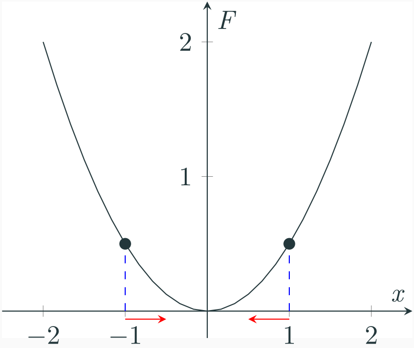

Figure 5: The function $F(x) = x^2/2$. Moving in the direction of $-F' = -x$ decreases F. The steepness of F is quantified by $|F'|$.

Moving in the direction of $-\nabla F$

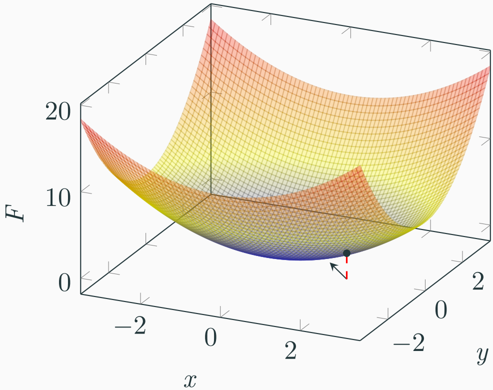

Figure 6: The function $F(x, y) = 0.5x^2 + y^2$ and the negative gradient direction at $(x, y) = (2, -1)$.