EE M146 — Discussion 2
Introduction to Machine Learning — Friday, April 12, 2019
Instructor: Lara Dolecek
TA: Zehui (Alex) Chen, Ruiyi (John) Wu

1. Matrix calculus review

   (a) Gradient of differentiable function $f : \mathbb{R}^n \to \mathbb{R}$:
   $\nabla f(x) = \left[\frac{\partial}{\partial x_1} f(x), \frac{\partial}{\partial x_2} f(x), \cdots, \frac{\partial}{\partial x_n} f(x)\right]^T$.

   - $\nabla_w(w^T b)$

   $$= \begin{bmatrix} \vdots \\ \frac{\partial \sum w_i b_i}{\partial w_i} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ b_i \\ \vdots \end{bmatrix} = b$$

   - $\nabla_w(\|w\|^2)$

   $$\frac{\partial \|w\|^2}{\partial w_i} = \frac{\partial w_1^2 + \cdots + w_i^2 + \cdots + w_n^2}{\partial w_i} = 2w_i$$

   - $\nabla_w(w^T A w)$ $= Aw + A^T w$

   $$\frac{\partial w^T A w}{\partial w_i} = \frac{\partial \sum_j \sum_k w_j A_{jk} w_k}{\partial w_i} = \frac{\partial w_i \sum_k w_k A_{ik}}{\partial w_i} + \frac{\partial w_i \sum_j w_j A_{ji}}{\partial w_i}$$

   $$= A_{(i,:)} w + A^T_{(:,i)} w$$

   - $\nabla_w(w^T X^T X w) =$ $2X^T X w$ using the previous result.

   (b) Jacobian/derivative matrix of differentiable function $f : \mathbb{R}^n \to \mathbb{R}^m$:

   $$\boldsymbol{J} = \begin{bmatrix} \nabla f_1(x)^T \\ \nabla f_1(x)^T \\ \vdots \\ \nabla f_m(x)^T \end{bmatrix}, \boldsymbol{J}_{ij} = \frac{\partial f_i}{\partial x_j}$$

   - Ax

   $$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} \quad J_{Ax} = \begin{bmatrix} \nabla a_1^T x \\ \vdots \\ \nabla a_m^T x \end{bmatrix} = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} = A$$

   $$\underset{m\times 1}{\Delta Ax} = \underset{m\times n}{J}\ \underset{n\times 1}{\Delta x}$$

- Example: transformation from polar (r, θ) to Cartesian coordinates (x, y): $x = r\cos(\theta), y = r\sin(\theta)$.

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{bmatrix} \begin{bmatrix} \Delta r \\ \Delta \theta \end{bmatrix} \qquad J = \begin{bmatrix} \cos\theta & -r\sin(\theta) \\ \sin\theta & r\cos(\theta) \end{bmatrix}$$

(c) Hessian matrix for twice differentiable function $f : \mathbb{R}^n \to \mathbb{R}$:
$\nabla^2 f(x)_{ij} = \frac{\partial^2}{\partial x_i \partial x_j} f(x)$.
The Hessian matrix is also the derivative matrix $\boldsymbol{J}$ of the gradient $\nabla f(x)$.

- Affine function $f(x) = a^T x + b$.

$$\nabla f(x) = a \qquad \nabla^2 f(x) = 0$$

- Least squares cost: $\|Ax - b\|^2$.

$$\nabla f(x) = 2A^T Ax - 2A^T b \qquad \nabla^2 f(x) = 2A^T A$$

- Example: $4x_1^2 + 4x_1x_2 + x_2^2 + 10x_1 + 9x_2$

$$\nabla f(x) = \begin{bmatrix} 8x_1 + 4x_2 + 10 \\ 4x_1 + 2x_2 + 9 \end{bmatrix} \qquad \nabla^2 f(x) = \begin{bmatrix} 8 & 4 \\ 4 & 2 \end{bmatrix}$$

2. Show that for a linearly separable data set, the maximum likelihood solution for the logistic regression model is obtained by finding a vector w whose decision boundary $w^T x = 0$ separates the classes and then taking the magnitude of w to infinity.

   Sketch of solution: If the dataset is linearly separable, then we can find w that for all points x_n belongs to class C_1, $w^T x_n > 0$; for all points x_m belongs to class C_2, $w^T x_m < 0$. According to the assumption of logistic regression, if we allow $|w| \to \infty$, for x_n belongs to C_1, $P(C_1|x_n, w) = \sigma(w^T x_n) \to 1$; for x_m belongs to C_2, $P(C_2|x_m, w) = 1 - \sigma(w^T x_m) \to 1$. This would maximize every term in the likelihood function and is therefore the ML solution.

   Hence, for a linearly separable dataset, the learning process may prefer to make $|w| \to \infty$ and use the linear boundary to label the datasets, which can cause severe over-fitting problem.

3. In class, we provided a probabilistic interpretation of ordinary least squares. We now try to provide a probabilistic interpretation of the weighted linear regression. Consider a model where each of the N samples is independently drawn according to a normal distribution

$$P(y_n|x_n, w) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{(y_n - w^T x_n)^2}{2\sigma_n^2}\right).$$

In this model, each y_n is drawn from a normal distribution with mean $w^T x_n$ and variance σ_n^2. The σ_n^2 are **known**. Write the log likelihood of this model as a function of w. Show that finding the maximum likelihood estimate of w leads to the same answer as solving a weighted linear regression. How do σ_n^2 relate to α_n?

Sketch of solution:

$$
\begin{aligned}
\text{argmax}_w \prod_{n=1}^{N} P(y_n|x_n, w) &= \text{argmax}_w \left(const - \frac{1}{2} \sum_{n=1}^{N} \frac{(y_n - w^T x_n)^2}{\sigma_n^2} \right) \\
&= \text{argmin}_w \sum_{n=1}^{N} \frac{(y_n - w^T x_n)^2}{\sigma_n^2}.
\end{aligned}
$$

This is the identical objective as $J(w)$ for weighted least squares with $\alpha_n = 1/\sigma_n^2$.