Machine Learning: Problem Set 1

Justin D. Thomas

April 13, 2013

1. Newton's method for computing least squares

(a) Find the Hessian of the cost function $J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(\theta^\top x^{(i)} - y^{(i)})^2$.

Solution to 1-(a):

Note that $x^{(i)} \in \mathbb{R}^N$, where N is the number of features. For vocabulary practice m is the number of training samples. We have $\theta^\top x^{(i)} \in \mathbb{R}$ and $y^{(i)} \in \mathbb{R}$. The function J goes from $\mathbb{R}^N$ to $\mathbb{R}$. So the Hessian of J is an $N \times N$ symmetric matrix of functions $\mathbb{R}^N \to \mathbb{R}$. In fact, we will see that the Hessian of J is an $N \times N$ matrix of constant functions.

By the chain rule, we compute

$$\partial J/\partial\theta_j = \sum_i x_j^{(i)}(\theta^\top x^{(i)} - y^{(i)}). \tag{1}$$

Thus the Hessian of J is

$$H(J)_{jk} = \sum_{i=1}^{m} x_j^{(i)} x_k^{(i)}, \tag{2}$$

which does not depend on θ, so is a matrix of constant functions, as claimed.

(b) Show that the first iteration of Newton's method gives us

$$\theta^\star = (X^\top X)^{-1} X^\top \vec{y},$$

the solution to the least squares problem.

Solution to 1-(b):

By definition, X is the matrix whose i^{th} row vector is $x^{(i)}$. Thus $X_{ij} = x_j^{(i)}$ and

$$(X^\top X)_{jk} = \sum_i X_{ji}^\top X_{ik} = \sum_i x_j^{(i)} x_k^{(i)} = H(J)_{jk}, \tag{3}$$

where the last equality is (2). We apply Newton's method to find a zero of ∇J with initial guess $\theta_0 = 0$. Recall that since $J \colon \mathbb{R}^N \to \mathbb{R}$ we have $\nabla J \colon \mathbb{R}^N \to \mathbb{R}^N$. We write ∇J as an N-dimensional column vector of $\mathbb{R}$-valued functions of N variables. In particular, the j^{th} row of ∇J is $\nabla_j J \coloneqq \partial J/\partial \theta_j$. By definition of Newton's method, we have

$$\theta_1 = \theta_0 - H(J)^{-1}(\nabla J)(\theta_0). \tag{4}$$

In this description θ_i is a column vector in $\mathbb{R}^N$. Since ∇J has dimension $N \times 1$ and $H(J)^{-1}$ has dimension $N \times N$, we see that the right hand side in the equation above is well-defined. Recall that ∇J is a function of θ. When $\theta = 0$ we see by equation (1) that

$$(\nabla J)(0) = -\sum_i x_j^{(i)} y^{(i)} = -\sum_i X_{ij} y^{(i)} = X^\top \vec{y}, \tag{5}$$

where, by definition, $\vec{y}$ is the $N \times 1$ vector whose i^{th} row is $y^{(i)}$, the i^{th} observed value in the training set. Putting $\theta_0 = 0$ into (4) and using equations (3) and (5), we have

$$\theta_1 = -(X^\top X)^{-1}(-X^\top \vec{y}) = (X^\top X)^{-1} X^\top \vec{y}, \tag{6}$$

as desired.

2. Locally-weighted logistic regression

Recall that logistic regression given by choosing $\theta \in \mathbb{R}^N$ to give the maximum likelihood of the sample set with the following prediction function

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^\top x}}. \tag{7}$$

In this model, $h_\theta(x)$ is predicting y, which takes values in $\{0, 1\}$. Given θ, then for each sample i the value

$$L_i(\theta) = h_\theta(x^{(i)})^{y^{(i)}}(1 - h_\theta(x^{(i)}))^{1-y^{(i)}}$$

is close to 1 if $h_\theta(x^{(i)})$ is close to $y^{(i)}$. Thus the product $L(\theta) = \prod_i L_i(\theta)$ is close to 1 if h_θ is a good predictor of $y^{(i)}$ given $x^{(i)}$ for all i. We think of this product as the likelihood of the sample $\{(x^{(i)}, y^{(i)}) \mid i = 1, \ldots, n\}$ when the parameter θ is given and h_θ is used to predict $y^{(i)}$ as a function of $x^{(i)}$. We want to maximize $L(\theta)$, but it is easier and equivalent to maximize $\ell(\theta) \coloneqq \log L(\theta)$.

$$\ell(\theta) = \sum_{i=1}^m y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) = \sum_{i=1}^m \ell_i(\theta) \tag{8}$$

The log likelihood for logistic regression.

Now we localize this around $x \in \mathbb{R}^N$ using the weight function

$$w^{(i)}(x) = \exp(-\left|x - x^{(i)}\right|^2 / (2\tau^2)), \tag{9}$$

where τ is a constant parameter. This gives us

$$\ell(\theta, x) = \sum_{i=1}^{m} w^{(i)}(x)\ell_i(\theta) \tag{10}$$

Locally-weighted log likelihood linear regression.

For reasons I don't understand, Newton's method does not work well for any (or some?) value(s) of x when finding maxima of the function $\theta \mapsto \ell(\theta, x)$. We can fix this by adding a linear term to $\partial_\theta \ell(\theta, x)$, or a θ-quadratic term to $\ell(\theta, x)$. We take a quadratic term of the form $(-\lambda/2)\left|\theta\right|^2$, or $\lambda\theta^\top\theta$, where λ is very small, say $\lambda = 0.0001$. The addition of a linear term to a function will adjust the critical points, but when the linear term is small, they are not too far off. Our adjusted likelihood function for locally weighted linear regression is

$$\ell'(\theta, x) = -\frac{\lambda}{2}\theta^\top\theta + \ell(\theta, x) \tag{11}$$

Adjusted log likelihood for locally-weighted linear regression.

Now we compute $\partial_\theta \ell$ (really the gradient). Clearly $\partial_\theta(-\lambda/2)\left|\theta\right|^2 = -\lambda\theta$. Also $\partial_\theta \ell(\theta, x) = \sum_i w^{(i)}(x)\partial_\theta \ell_i(\theta)$. So we need to compute $\partial_\theta \ell_i(\theta)$. We use

$$\begin{aligned} \partial_{\theta_j} h_\theta(x) &= \partial_{\theta_j} \frac{1}{1 + e^{-\theta^\top x}} \\ &= \frac{x_j e^{-\theta^\top x}}{(1 + e^{-\theta^\top x})^2} \\ &= x_j h_\theta(x)(1 - h_\theta(x)) \end{aligned} \tag{12}$$

Thus

$$\begin{aligned} \partial_{\theta_j} \log h_\theta(x^{(i)}) &= x_j^{(i)}(1 - h_\theta(x^{(i)})), \\ \partial_{\theta_j} \log(1 - h_\theta(x^{(i)})) &= -x_j^{(i)} h_\theta(x^{(i)}). \end{aligned}$$

Now if we set $z \in \mathbb{R}^m$ to be the column vector with i^{th} entry, z_i, given by

$w^{(i)}(y^{(i)} - h_\theta(x^{(i)}))$ we have

$$\begin{aligned}
\partial_{\theta_j}\ell'(\theta, x) &= -\lambda\theta_j + \sum_{i=1}^{m} w^{(i)}(x)x_j^{(i)}(y^{(i)}(1 - h_\theta(x^{(i)})) - (1 - y^{(i)})h_\theta(x^{(i)})) \\
&= -\lambda\theta_j + \sum_{i=1}^{m} w^{(i)}(x)x_j^{(i)}(y^{(i)} - h_\theta(x^{(i)})) \\
&= -\lambda\theta_j + \sum_{i=1}^{m} X_{ji}^\top z_i \\
&= -\lambda\theta_j + (X^\top z)_j
\end{aligned}$$

Dropping j from the notation we get an equality of functions $\mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$,

$$\nabla_\theta \ell'(\theta, x) = -\lambda\theta + X^\top z \tag{13}$$

Going further, we get $\partial_{jk}\ell' = \partial_j(X^\top z)_k - \lambda\delta_{jk}$. Note that $\partial_j z_i = -\partial_j h_\theta(x^{(i)}))$. By (12), $\partial_j h_\theta(x^{(i)}) = x_j^{(i)} h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))$. Together, these equations give

$$\begin{aligned}
H\ell'(\theta, x)_{jk} &= \partial_{jk}\ell'(\theta, x) \\
&= -\lambda\delta_{jk} + \sum_i X_{ki}^\top \partial_j z_i \\
&= (-\lambda I)_{jk} + \sum_i X_{ki}^\top x_j^{(i)} h_\theta(x^{(i)})(1 - h_\theta(x^{(i)})) \\
&= (-\lambda I)_{jk} + \sum_{i,l} X_{ki}^\top h_\theta(x^{(l)})(1 - h_\theta(x^{(l)}))\delta_{il} X_{lj} \\
&= (-\lambda I + X^\top D X)_{jk},
\end{aligned} \tag{14}$$

where $D_{il} = h_\theta(x^{(l)})(1 - h_\theta(x^{(l)}))\delta_{il}$ is a diagonal $m \times m$ matrix.

(a) Implement the Newton-Raphson algorithm for optimizing $\ell(\theta)$ for a new query point x, and use this to predict the class of x.

Solution to 2-(a):

Recall that the Newton-Raphson algorithm is just another name for the update rule

$$\theta := \theta - H^{-1}\nabla_\theta \ell(\theta),$$

used for finding the critical points of $\ell(\theta)$.

The code for this problem can be found in the IPython notebook in the same directory as this document, `lwlr.ipynb`.

3. Multivariate Least Squares Given a training set $\mathbb{R}^n \leftarrow S \to \mathbb{R}^p$, find a linear function $L\colon \mathbb{R}^n \to \mathbb{R}^p$ best approximating this training set. More concretely, $S = \{1, \ldots, m\}$ and for each $i \in S$ we have $x^{(i)} \in \mathbb{R}^n$ and $y^{(i)} \in \mathbb{R}^p$. The linear function L is usually thought of as an $n \times p$ matrix θ. That is, $L(x) = \theta^\top x$. We want to find L such that the sum over all $i \in S$ of the distances from $L(x^{(i)})$ to $y^{(i)}$ is minimal. That is, we want θ minimizing the cost function

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{p} \left((\theta^\top x^{(i)})_j - y_j^{(i)} \right)^2 . \tag{15}$$

(a) Write $J(\theta)$ from equation (15) in matrix-vector notation.

Solution to 3-(a):

Recall that X is the $m \times n$ matrix $X_{ij} = x_j^{(i)}$. So $X^\top$ has columns given by the $x^{(i)}$. Thus $\theta^\top X^\top$ has m columns given by the $\theta^\top x^{(i)} \in \mathbb{R}^p$. Let $Y^\top$ be the $p \times m$ matrix having columns given by the $y^{(i)}$, then $J(\theta)$ is half the sum of the norms of the column vectors of the $p \times m$ matrix $\theta^\top X^\top - Y^\top$. Given any matrix A, the square matrix $A^\top A$ has diagonal entries equal to the squared norms of the column vectors of A. Thus the trace of $A^\top A$, denoted $\operatorname{tr}(A^\top A)$, is the sum of the squared norms of the column vectors of A. We conclude that

$$J(\theta) = \frac{1}{2} \operatorname{tr}((X\theta - Y)(\theta^\top X^\top - Y^\top)) = \frac{1}{2} |X\theta - Y|^2 \tag{16}$$

(b) Find the closed form solution for θ which minimizes $J(\theta)$. This is the equivalent to the normal equations for the multivariate case.

Solution to 3-(b):

In coordinates,

$$\begin{aligned} J(\theta) &= \frac{1}{2} \sum_{i,j} ((X\theta - Y)_{ij})^2 \\ &= \frac{1}{2} \sum_{i,j} \left(-Y_{ij} + \sum_p X_{ip}\theta_{pj} \right)^2 \end{aligned}$$

Let $\partial_{\theta_{ij}} = \partial_{ij}$, then

$$\begin{aligned}
\partial_{kl} J(\theta) &= \sum_{i,j} \left(-Y_{ij} + \sum_p X_{ip}\theta_{pj} \right) \left(\sum_p X_{ip} \delta_k^p \delta_l^j \right) \\
&= \sum_{i,j} (-Y_{ij} + \sum_p X_{ip}\theta_{pj}) X_{ik} \delta_l^j \\
&= \sum_i \left(-X_{ik} Y_{il} + X_{ik} \sum_p X_{ip}\theta_{pl} \right) \\
&= (-X^\top Y + X^\top X\theta)_{kl}
\end{aligned}$$

In invariant notation, we use the facts $\nabla_\theta \operatorname{tr}(\theta A) = A^\top$ and $\nabla_\theta \operatorname{tr}(\theta^\top B) = B$. It helps me to understand these formulas in terms of the derivative $df\colon TM \to T\mathbb{R}$ of a function of manifolds $f\colon M \to \mathbb{R}$. In this case, we are differentiating with respect to θ, so we should have $\theta \in T_\eta M$ for some manifold M and some η. Therefore let us put $M = M_{n\times p}$, then η is any point of M since $\theta \in M_{n\times p} \simeq T_\eta M_{n\times p}$ for all η. Let $f = \operatorname{tr} \circ R_A \colon \eta \mapsto \eta A \mapsto \operatorname{tr}(\eta A)$. Then, by definition, $\nabla_\theta \operatorname{tr}(\theta A)$ is the $n \times p$ matrix of functions on $M_{n\times p}$ whose (i,j) entry is given by $\eta \mapsto (df)_\eta(E_{ij})$, where E_{ij} is the matrix whose (k,l) entry is $\delta_i^k \delta_j^l$.

Let us coordinatize $M_{n\times p}$ by $(\eta_{ij})\colon \mathbb{R}^{n\times p} \to M_{n\times p}$. We have $T\mathbb{R}^{n\times p} = \mathbb{R}^{n\times p} \times \mathbb{R}^{n\times p}$ and, in this coordinate system, $df\colon \mathbb{R}^{n\times p} \times \mathbb{R}^{n\times p} \to \mathbb{R}$, $df(\eta, E) = (df)_\eta(E)$, which is $\lim_{\epsilon\to 0} \frac{1}{\epsilon}(f(\eta + \epsilon E) - f(\eta))$. Since f is linear, this is just $f(E) = \operatorname{tr}(EA)$. Let's compute $\operatorname{tr}(E_{ij}A)$. We have $(E_{ij}A)_{kl} = \sum_m (E_{ij})_{km} A_{ml} = \delta_i^k A_{jl}$. Therefore $\operatorname{tr}(E_{ij}A) = \sum_m (E_{ij}A)_{mm} = A_{ji} = (A^\top)_{ij}$. Thus $(df)_\eta(E)$ is independent of η. It is is just the $n \times p$ matrix $A^\top$.

If $f\colon M_{n\times p} \to \mathbb{R}$, then $f \circ ()^\top$ has gradient $(\nabla f)^\top$. Indeed,

$$\begin{aligned}
(\partial_{\theta_{ij}} f \circ ()^\top)(\theta) &= \lim_{\epsilon\to 0} \frac{1}{\epsilon}(f((\theta + \epsilon E_{ij})^\top) - f(\theta^\top)) \\
&= \lim_{\epsilon\to 0} \frac{1}{\epsilon}(f(\theta^\top + \epsilon E_{ji}) - f(\theta^\top)) \\
&= (\partial_{\theta_{ji}} f)(\theta^\top).
\end{aligned}$$

This implies that $\nabla_\theta \operatorname{tr}(\theta^\top B) = (B^\top)^\top = B$. Note that the $\theta^\top$ in the final line above does not affect us here because for $f(\theta) = \operatorname{tr}(\theta A)$, $\partial_{\theta_{ij}} f$ is independent of θ.

We apply the product rule to the function $\theta \mapsto \operatorname{tr}(X\theta\theta^\top X^\top)$ by permuting θ and $\theta^\top$ to the front. Recall that if A and B are matrices so that AB is defined and is square, then the same holds for BA and $\operatorname{tr}(AB) = \operatorname{tr}(BA)$. In the expression $X\theta\theta^\top X^\top$ we can take $A = X$ and $B = \theta\theta^\top X^\top$. Then AB is defined and AB is square, so we get $\operatorname{tr}(X\theta\theta^\top X^\top) = \operatorname{tr}(\theta\theta^\top X^\top X)$.

Similarly, we have $\operatorname{tr}(\theta\theta^\top X^\top X) = \operatorname{tr}(\theta^\top X^\top X\theta)$. Using these facts, we apply the product rule to compute

$$\begin{aligned}
\nabla_\theta \operatorname{tr}(X\theta\theta^\top X^\top) &= \nabla_\theta \operatorname{tr}(\theta\theta^\top X^\top X) + \nabla_\theta \operatorname{tr}(\theta^\top X^\top X\theta) \\
&= (\theta^\top X^\top X)^\top + X^\top X\theta \\
&= 2X^\top X\theta. \qquad (17)
\end{aligned}$$

Finally (16) and (17) give can be used to compute $\nabla_\theta J(\theta)$ in invariant notation.

$$\begin{aligned}
\nabla_\theta J(\theta) &= \frac{1}{2}\nabla_\theta \left(\operatorname{tr}(X\theta\theta^\top X^\top) - 2\operatorname{tr}(X\theta Y^\top) + \operatorname{tr}(YY^\top)\right) \\
&= \frac{1}{2}(2X^\top X\theta - 2X^\top Y) \\
&= X^\top X\theta - X^\top Y \qquad (18)
\end{aligned}$$

Finally, setting $\nabla_\theta J(\theta) = 0$, we get $X^\top X\theta = XY$. Recall that the square root of $\det(X^\top X)$ is the volume-change factor of the linear map $X\colon \mathbb{R}^m \to \mathbb{R}^n$. So as long as we have enough points $x^{(i)} \in \mathbb{R}^n$ so that the span of $\{x^{(i)} \mid i = 1, \ldots, m\}$ is all of $\mathbb{R}^n$, then $\det(X^\top X)$ must be non-zero. If all of the sample points lie in some k-dimensional subspace of $\mathbb{R}^n$ with $k < n$, we just refactor the data to replace n by k. Thus, we can assume $X^\top X$ is invertible without loss of generality. This means we can solve for θ. We have $\theta = (X^\top X)^{-1}X^\top Y$.

4. Naive Bayes

Let W be a finite set. Let a feature vector x be a map $W \to \{0,1\}$. Let an output vector y be a point of $\{0,1\}$. Given a finite set $S = \{1, \ldots, m\}$ and training data $\operatorname{map}(W, \{0,1\}) \leftarrow S \to \{0,1\}$, we seek a function $\operatorname{map}(W, \{0,1\}) \to \{0,1\}$ approximating the training data as closely as possible.

Concretely, W is a list of words $W = \{\textit{apple}, \textit{boat}, \textit{jet}, \ldots\}$, x is a subset of W given by an email, and y is a classification of x as *spam* or *not spam*.

Given a map $P\colon \operatorname{map}(W, \{0,1\}) \times \{0,1\} \to \mathbb{R}$ we get a function

$$f_P\colon \operatorname{map}(W, \{0,1\}) \to \{0,1\} \quad f_P(x) = 1 \iff P(x,1) \geq P(x,0).$$

Since f_P is invariant under transformations $P \mapsto aP + b$, where $a, b \in \mathbb{R}$, $a > 0$, we may as well assume $P \geq 0$ and the integral of P is 1. That is, assume P is a probability distribution. We will use the lower case p to refer to a probability distribution on $\operatorname{map}(W, \{0,1\}) \times \{0,1\}$.

We have $p(x,y) = p(x|y)p(y)$. We assume $p(y)$ is binomially distributed with $\phi_y := p(y = 1)$. In addition, we assume $p(x|y) = \prod_j p(x(j) = 1|y)$. Thus we are assuming that, given y, the value of x at $j \in W$ is independent of the value at $j' \in W$, where $j' \neq j$. This is a naive assumption, thus the word *naive* in the name of this model. For instance in the spam email classifier example we are assuming that if we know an email is spam and the word *win* appears, this has no effect on the probability that *money* appears. This assumption seems unlikely to be true, but the algorithm still returns reasonable classifications.

This assumption means we only need to specify probabilities $\phi_y, \phi_{j|y=0}$, and $\phi_{j|i} \in [0,1]$ to determine p_ϕ, where $\phi_y = p_\phi(y=1)$, $\phi_{j|i} = p_\phi(x(j) = 1|y=i)$.

(a) Reconstruct p_ϕ from ϕ and reconstruct the joint likelihood function

$$\ell(\phi) = \log \prod_{i \in S} p_\phi(x^{(i)}, y^{(i)}).$$

Solution to 4-(a):

Let $p = p_\phi$. We have

$$\begin{aligned} p(x,y) &= p(x|y)p(y) \\ &= p(x|y)\phi_y^y(1-\phi_y)^{1-y} \\ &= p(x|y=0)^{1-y}p(x|y=1)^y\phi_y^y(1-\phi_y)^{1-y} \\ &= \phi_y^y(1-\phi_y)^{1-y}\prod_{j\in W}\phi_{j|0}^{x_j(1-y)}(1-\phi_{j|0})^{(1-x_j)(1-y)}\phi_{j|1}^{x_jy}(1-\phi_{j|1})^{(1-x_j)y} \end{aligned}$$

Now compute

$$\begin{aligned} \ell(\phi) &= \sum_{i\in S} y^{(i)}\log\phi_y + (1-y^{(i)})\log(1-\phi_y) \\ &\quad + \sum_{i\in S}\sum_{j\in W} x_j^{(i)}(1-y^{(i)})\log(\phi_{j|0}) + (1-x_j^{(i)})(1-y^{(i)})\log(1-\phi_{j|0}) \\ &\quad + \sum_{i\in S}\sum_{j\in W} x_j^{(i)}y^{(i)}\log(\phi_{j|1}) + (1-x_j^{(i)})y^{(i)}\log(1-\phi_{j|1}) \end{aligned}$$

(b) Show that the parameters which maximize the likelihood function are the same as those given in the lecture notes. (We will just show the answer once since it is long.)

Compute some derivatives.

$$\begin{aligned}
\partial_{\phi_y}\ell(\phi) &= \sum_{i\in S} y^{(i)}/\phi_y - (1-y^{(i)})/(1-\phi_y) \\
\partial_{\phi_{j|0}}\ell(\phi) &= \sum_{i\in S} x_j^{(i)}(1-y^{(i)})/\phi_{j|0} - (1-x_j^{(i)})(1-y^{(i)})/(1-\phi_{j|0}) \\
\partial_{\phi_{j|1}}\ell(\phi) &= \sum_{i\in S} x_j^{(i)}y^{(i)}/\phi_{j|1} - (1-x_j^{(i)})y^{(i)}/(1-\phi_{j|1})
\end{aligned}$$

Setting $\partial_{\phi_y} = 0$, we get $0 = (1-\phi_y)(\sum y^{(i)}) - \phi_y \sum(1-y^{(i)})$, which simplifies to $0 = (\sum y^{(i)}) - \phi_y\,|S|$. Thus $\phi_y = |S_1|\,/\,|S|$, where $S_k = \{i \in S \mid y^{(i)} = k\}$. That is, ϕ_y is the number of i in the training set S classified as $y = 1$.

Similarly, setting $\partial_{\phi_{j|0}}\ell(\phi) = 0$, we get

$$\begin{aligned}
0 &= (1-\phi_{j|0})\left(\sum x_j^{(i)}(1-y^{(i)})\right) - \phi_{j|0}\sum(1-x_j^{(i)})(1-y^{(i)}) \\
&= \left(\sum x_j^{(i)}(1-y^{(i)})\right) - \phi_{j|0}\sum(1-y^{(i)})
\end{aligned}$$

Solving for $\phi_{j|0}$ we get $\sum_{i\in S_0} x_j^{(i)}/\,|S_0|$. This is just the fraction of the $y = 0$ training set where $x(j)$ takes the value 1. In email parlance, it is the fraction of non-spam emails in which the j^{th} word in W appears at least once.

A reasonable guess is the $\phi_{j|1}$ is the fraction of spam emails containing the j^{th} word at least once. This is $\sum_{i\in S_1} x_j^{(i)}/\,|S_1|$. We can derive this by setting $\partial_{\phi_{j|1}} = 0$, then

$$\begin{aligned}
0 &= (1-\phi_{j|1})(\sum x_j^{(i)}y^{(i)}) - \phi_{j|1}\sum(1-x_j^{(i)})y^{(i)} \\
&= \sum x_j^{(i)}y^{(i)} - \phi_{j|1}\sum y^{(i)} \\
\phi_{j|1} &= (\sum_{i\in S_1} x_j^{(i)})/\,|S_1|
\end{aligned}$$

In the notation $1\{\text{true}\} = 1$ and $1\{\text{false}\} = 0$ and $S = \{1,\ldots,m\}$ we put the formulas as they occur in the notes. In addition, if we define $S^j := \{i \in S \mid x_j^{(i)} = 1\}$, then we have significantly abbreviated formulas.

$$\begin{aligned}
\phi_y &= \frac{\sum_{i=1}^m 1\{y^{(i)} = 1\}}{m} = \frac{|S_1|}{|S|} \\
\phi_{j|k} &= \frac{\sum_{i=1}^m 1\{x^{(i)} = 1 \wedge y^{(i)} = k\}}{\sum_{i=1}^m 1\{y^{(i)} = k\}} = \frac{|S^j \cap S_k|}{|S_k|}
\end{aligned}$$

(c) Show that the naive Bayes algorithm is a linear classifier. That is, given $x \in \mathbb{R}^n$, show that there exists $\theta \in \mathbb{R}^{n+1}$ such that

$$p(y=1|x) \geq p(y=0|x) \iff \theta^\top \begin{pmatrix} 1 \\ x \end{pmatrix} \geq 0$$

Bayes rule says

$$
\begin{aligned}
p(y=k|x) &= \frac{p(x|y=k)p(y=k)}{p(x)} \\
&= \frac{\phi_y^k(1-\phi_y)^{1-k}\prod_j \phi_{j|k}^{x_j}(1-\phi_{j|k})^{1-x_j}}{p(x)}
\end{aligned}
$$

Clearly $p(y=1|x) \geq p(y=0|x)$ if and only if $\log p(y=1|x) \geq \log p(y=0|x)$ since $p \geq 0$ and log is an increasing function. Let us compute the difference of these logarithms.

$$
\begin{aligned}
&\log p(y=1|x) - \log p(y=0|x) = \\
&\qquad = \log \phi_y + \sum_j (x_j \log \phi_{j|1} + (1-x_j)\log(1-\phi_{j|1})) \\
&\qquad\quad - \log(1-\phi_y) - \sum_j (x_j \log \phi_{j|0} + (1-x_j)\log(1-\phi_{j|0})) \\
&\qquad = \log\left(\frac{\phi_y \prod_j (1-\phi_{j|1})}{(1-\phi_y)\prod_j(1-\phi_{j|0})}\right) + \sum_j \log\left(\frac{\phi_{j|1}(1-\phi_{j|0})}{(1-\phi_{j|1})\phi_{j|0}}\right) x_j \\
&\qquad = \theta_0 \cdot 1 + \sum_j \theta_j x_j = \theta^\top \begin{pmatrix} 1 \\ x \end{pmatrix},
\end{aligned}
$$

where $\theta_0 = \log(\phi_y \prod_j (1-\phi_{j|1})) - \log((1-\phi_y)\prod_j(1-\phi_{j|0}))$ and $\theta_j = \log(\phi_{j|1}(1-\phi_{j|0})) - \log((1-\phi_{j|1})\phi_{j|0})$. This completes the proof that naive Bayes is a linear classifier.

5. Exponential family and the geometric distribution

(a) Consider the geometric distribution parameterized by ϕ:

$$p(y;\phi) = (1-\phi)^{y-1}\phi, y = 1, 2, 3, \ldots$$

Show that the geometric distribution is in the exponential family, and give $b(y)$, η, $T(y)$, and $a(\eta)$.

Solution to 5-(a):

Recall that the exponential distribution with parameter η and functions $b(y), T(y), a(\eta)$ is

$$p(y;\eta) = b(y)\exp(\eta^\top T(y) - a(\eta)).$$

To write $(1-\phi)^{y-1}\phi$ as an exponential, we take logs. We have $p(y;\eta) = \exp((y-1)\log(1-\phi) + \log\phi)$. Inside the exponential, we isolate the y-dependence, which is $y\log(1-\phi)$. This implies we should guess $T(y) = y$

and $\eta = \log(1-\phi)$. This implies we need to write the remaining part of the exponential as a function of $\log(1-\phi)$. The remaining piece is $\log(\phi/(1-\phi))$. We can write this as $\log(1-e^{\eta}) - \eta$. This is what we take $-a(\eta)$ to be. We set $b(y) = 1$. In summary,

$$\eta = \log(1-\phi) \quad T(y) = y \quad a(\eta) = \eta - \log(1-e^{\eta}) \quad b(y) = y,$$

gives

$$b(y)\exp(\eta^{\top}T(y) - a(\eta)) = (1-\phi)^{y-1}\phi.$$

(b) Consider performing regression using a GLM with a geometric response variable. What is the canonical response function for the family?

Solution to 5-(b):

Recall that the target variable y in a generalized linear model (GLM) is also called the response variable. The response function is $\eta \mapsto E[T(y);\eta]$, the expectation value of $T(y)$ parameterized by η. In the case of the geometric distribution, we computed $T(y) = y$ in the previous part of this problem. Thus we need to compute $\sum_{y=1,2,3,\ldots}(1-\phi)^{y-1}\phi y$. This is $\sum_{y\geq 1} -\partial_{\phi}((1-\phi)^{y}\phi) + (1-\phi)^{y}$. The geometric series $\sum_{y\geq 0}(1-\phi)^{y}$ converges to $1/\phi$. This implies

$$E[y;\phi] = -\partial_{\phi}(\phi(\frac{1}{\phi} - 1)) + \frac{1}{\phi} - 1 = \frac{1}{\phi}$$

Therefore $E[T(y);\eta] = E[y;\phi(\eta)] = E[y;1-e^{\eta}] = 1/(1-e^{\eta})$ is the canonical response function for this model.

(c) For a training set $\{(x^{(i)}, y^{(i)}); i = 1, \ldots m\}$, let the log-likelihood of an example be $\log p(y^{(i)}|x^{(i)};\theta)$. By taking the derivative of the log-likelihood with respect to θ_j, derive the stochastic gradient ascent rule for learning using a GLM with geometric responses y and the canonical response function.

Solution to 5-(c):

Recall that in generalized linear models (GLM's) θ is related to the GLM parameters by setting $\eta = \theta^{\top}x$. With this we have $\phi = 1 - e^{\theta^{\top}x}$ so $p(y|x;\theta) = (1-\phi)^{y-1}\phi = e^{(y-1)\theta^{\top}x}(1-e^{\theta^{\top}x})$. Thus we think of θ as a more fundamental parameter, giving us a range of η's. Also, θ gives the prediction function $x \mapsto h_{\theta}(x) := E[y;\theta^{\top}x]$. Now the log-likelihood of the sample is

$$\ell(\theta) = \sum_{i}(y^{(i)} - 1)\theta^{\top}x^{(i)} + \log(1 - e^{\theta^{\top}x})$$

Now compute the gradient.

$$
\begin{aligned}
\partial_{\theta_j} \ell(\theta) &= \sum_i (y^{(i)} - 1) x_j^{(i)} + \frac{-x_j e^{\theta^\top x^{(i)}}}{1 - e^{\theta^\top x^{(i)}}} \\
&= \sum_i y^{(i)} x_j^{(i)} - \frac{x_j}{1 - e^{\theta^\top x}} \\
&= \sum_i y^{(i)} x_j^{(i)} - h_\theta(x^{(i)}) x_j^{(i)}
\end{aligned}
$$

Therefore the stochastic update rule for gradient ascent is

$$
\begin{aligned}
\theta_j &:= \theta_j + \alpha (y^{(i)} - h_\theta(x^{(i)})) x_j \\
&= \theta_j + \alpha \left(y^{(i)} - \frac{x_j^{(i)}}{1 - e^{\theta^\top x}} \right)
\end{aligned}
$$

where α is a small constant parameter indicating the speed of gradient ascent.