SDS 383D Exercises 1: Preliminaries

Tianqi Chen (tc34927)

1 Bayesian inference in simple conjugate families

We start with a few of the simplest building blocks for complex multivariate statistical models: the beta/binomial, normal, and inverse-gamma conjugate families.

(A) Suppose that we take independent observations $x_1, \ldots, x_N$ from a Bernoulli sampling model with unknown probability w. That is, the x_i are the results of flipping a coin with unknown bias. Suppose that w is given a Beta(a,b) prior distribution:

$$p(w) = \frac{\Gamma(a+b)}{\Gamma(a) \cdot \Gamma(b)} \; w^{a-1}(1-w)^{b-1} \,,$$

where $\Gamma(\cdot)$ denotes the Gamma function. Derive the posterior distribution $p(w \mid x_1, \ldots, x_N)$.[1]

(B) The probability density function (PDF) of a gamma random variable, $x \sim \mathrm{Ga}(a, b)$, is

$$p(x) = \frac{b^a}{\Gamma(a)} x^{a-1} \exp(-bx) \,.$$

Suppose that $x_1 \sim \mathrm{Ga}(a_1, 1)$ and that $x_2 \sim \mathrm{Ga}(a_2, 1)$. Define two new random variables $y_1 = x_1/(x_1 + x_2)$ and $y_2 = x_1 + x_2$. Find the joint density for (y_1, y_2) using a direct PDF transformation (and its Jacobian).[2] Use this to characterize the marginals $p(y_1)$ and $p(y_2)$, and propose a method that exploits this result to simulate beta random variables, assuming you have a source of gamma random variables.

(C) Suppose that we take independent observations $x_1, \ldots, x_N$ from a normal sampling model with unknown mean θ and *known* variance σ^2: $x_i \sim \mathrm{N}(\theta, \sigma^2)$. Suppose that θ is given a normal prior distribution with mean m and variance v. Derive the posterior distribution $p(\theta \mid x_1, \ldots, x_N)$.

(D) Suppose that we take independent observations $x_1, \ldots, x_N$ from a normal sampling model with *known* mean θ but *unknown* variance σ^2. (This seems even more artificial than the last, but is conceptually important.) To make this easier, we will re-express things in terms of the precision, or inverse variance $\omega = 1/\sigma^2$:

$$p(x_i \mid \theta, \omega) = \left(\frac{\omega}{2\pi}\right)^{1/2} \exp\left\{-\frac{\omega}{2}(x_i - \theta)^2\right\} \,.$$

[1] I offer two tips here that are quite general. (1) Your final expression will be cleaner if you reduce the data to a sufficient statistic. (2) Start off by ignoring normalization constants (that is, factors in the density function that do not depend upon the unknown parameter, and are only there to make the density integrate to 1.) At the end, re-instate these normalization constants based on the functional form of the density.

[2] Take care that you apply the important change-of-variable formula from basic probability. See, e.g., Section 1.2 of `http://www.stat.umn.edu/geyer/old/5102/n.pdf`.

Suppose that ω has a gamma prior with parameters a and b, implying that σ^2 has what is called an inverse-gamma prior.[3] Derive the posterior distribution $p(\omega \mid x_1, \ldots, x_N)$. Re-express this as a posterior for σ^2, the variance.

(E) Suppose that, as above, we take independent observations $x_1, \ldots, x_N$ from a normal sampling model with unknown, common mean θ. This time, however, each observation has its own idiosyncratic (but known) variance: $x_i \sim \mathrm{N}(\theta, \sigma_i^2)$. Suppose that θ is given a normal prior distribution with mean m and variance v. Derive the posterior distribution $p(\theta \mid x_1, \ldots, x_N)$. Express the posterior mean in a form that is clearly interpretable as a weighted average of the observations and the prior mean.

(F) Suppose that $(x \mid \omega) \sim \mathcal{N}(m, \omega^{-1})$, and that ω has a Gamma$(a/2, b/2)$ prior, with PDF defined as above. Show that the marginal distribution of x is Student's t with d degrees of freedom, center m, and scale parameter $(b/a)^{1/2}$. This is why the t distribution is often referred to as a *scale mixture of normals*.

Answer

(A) Write down the joint probability, we can see that

$$\begin{aligned} p(x_1, \ldots, w_N, w) &\propto w^{a-1}(1-w)^{b-1}\Pi_{i=1}^{N} w^{x_i}(1-w)^{1-x_i} \\ &= w^{a+\sum_{i=1}^{N} x_i}(1-w)^{N-\sum_{i=1}^{N} x_i}\,. \end{aligned}$$

Therefore, $w \mid x_1, \ldots, x_N \sim Beta(a+s_N, b+N-s_N)$, where $s_N = \sum_{i=1}^{N} x_i$. The corresponding density function is

$$p(w|x_1, \ldots, x_N) = \frac{\Gamma(a+b+N)}{\Gamma(a+s_N)\cdot\Gamma(b+N-s_N)} w^{a+s_N-1}(1-w)^{b+N-S_N-1}\,.$$

(B) The Jacobian matrix is

$$\frac{\partial(x_1, x_2)}{\partial(y_1, y_2)} = \begin{bmatrix} y_2 & y_1 \\ -y_2 & 1-y_1 \end{bmatrix}.$$

By change of variable theorem, we have

$$\begin{aligned} p(y_1, y_2) &= p(x_1(y_1, y_2), x_2(y_1, y_2)) \left| det\left(\frac{\partial(x_1, x_2)}{\partial(y_1, y_2)}\right)\right| \\ &= \frac{1}{\Gamma(a_1)\Gamma(a_2)}(y_1 y_2)^{a_1-1}(y_2 - y_1 y_2)^{a_2-1} exp(-y_2) y_2 \\ &= \frac{\Gamma(a_1+a_2)}{\Gamma(a_1-2)\Gamma(a_2)} y_1^{a_1-1}(1-y_1)^{a_2-1} \frac{1}{\Gamma(a_1+a_2)} y^{a_1+a_2-1} exp(-y_2) \end{aligned}$$

Notice, the joint density function, $p(y_1, y_2) = p(y_1)p(y_2)$. Therefore, y_1 and y_2 are independent, following $Beta(a_1, a_2)$ and $\Gamma(a_1 + a_2)$ respectively. A method for generating beta random

[3] Written $\sigma^2 \sim \mathrm{IG}(a, b)$.

variable from gamma random variables will be as follows: given two independent gamma random variables $x_1 \sim \Gamma(a_1)$ and $x_2 \sim \Gamma(a_2)$, $y = x_1/(x_1 + x_2)$ will follow a beta distribution, $Beta(a_1, a_2)$.

(C) The joint density is:

$$\begin{aligned} p(x_1, \ldots, x_N, \theta) &\propto exp\left(-\frac{1}{2v}(\theta - m)^2\right) \Pi_{i=1}^N exp\left(-\frac{1}{2\sigma^2}(x_i - \theta)^2\right) \\ &= exp\left(-\frac{1}{2v}(\theta - m)^2\right) exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^N (x_i - \theta)^2\right) \\ &\propto exp\left(-\frac{1}{2v}(\theta - m)^2 - \frac{N}{2\sigma^2}(\theta - \bar{x}_N)^2\right) \end{aligned}$$

where $\bar{x}_N = \sum_{i=1}^N x_i/N$.

Therefore, $\theta|x_1, \ldots, x_N \sim \mathcal{N}v_N m_N, v_N)$, where $m_N = \frac{m}{v} + \frac{N\bar{x}_N}{\sigma^2}, v_N = 1/(\frac{1}{v} + \frac{N}{\sigma^2})$.

(D) The joint density is:

$$\begin{aligned} p(x_1, \ldots, x_N, \omega) &\propto \omega^{a-1} exp(-b\omega) \Pi_{i=1}^N \omega^{1/2} exp\left(-\frac{\omega}{2}(x_i - \theta)^2\right) \\ &= \omega^{a+N/2-1} exp(-\omega(b + N\sigma_N^2/2)) \end{aligned}$$

where $\sigma_N^2 = \frac{\sum_{i=1}^N (x_i - \theta)^2}{N}$.

Therefore, $\omega|x_1, \ldots, x_N \sim \Gamma(a + N/2, b + N\sigma_N^2/2)$. By transformation $\sigma^2 = 1/\omega$, we have $\sigma^2|x_1, \ldots, x_N \sim Inverse - \Gamma(a + N/2 =: a_N, b + N\sigma_N^2/2 =: b_N)$.

$$p(\omega|x_1, \ldots, x_N) = \frac{b_N^{a_N}}{\Gamma(a_N)}(\sigma^2)^{-a_N - 1} exp(-b_N/\sigma^2)$$

(E) The joint density is:

$$\begin{aligned} p(x_1, \ldots, x_N, \theta) &\propto exp\left(-\frac{1}{2v}(\theta - m)^2\right) \Pi_{i=1}^N exp\left(-\frac{1}{2\sigma_i^2}(x_i - \theta)^2\right) \\ &= exp\left(-\frac{1}{2v}(\theta - m)^2\right) exp\left(-\frac{1}{2\sigma_i^2}\sum_{i=1}^N (x_i - \theta)^2\right) \\ &\propto exp\left(-\frac{1}{2v}(\theta - m)^2 - \frac{N}{2\sigma_N^2}(\theta - \tilde{x}_N)^2\right) \end{aligned}$$

where $\sigma_N^2 = \frac{N}{\sum_{i=1}^N 1/\sigma_i^2}, \tilde{x}_N = \frac{x_i/\sigma_i^2}{N/\sigma_N^2}$.

Similar to (C), $\theta|x_1, \ldots, x_N \sim \mathcal{N}\tilde{v}_N \tilde{m}_N, \tilde{v}_N)$, where $\tilde{m}_N = \frac{m}{v} + \frac{N\tilde{x}_N}{\sigma_N^2}, \tilde{v}_N = 1/(\frac{1}{v} + \frac{N}{\sigma_N^2})$.

(F) The joint density is:

$$p(x,\omega) = \frac{\omega^{1/2}}{\sqrt{2\pi}} exp\left(-\frac{\omega}{2}(x-m)^2\right) \frac{(b/2)^{a/2}}{\Gamma(a/2)} \omega^{a/2-1} exp(-b\omega/2)$$
$$= \frac{(b/2)^{a/2}}{\sqrt{2\pi}\Gamma(a/2)} \omega^{(a+1)/2-1} exp\left(-\left(b+(x-m)^2\right)\omega/2\right)$$

Integrate over ω, we have:

$$p(x) = \frac{(b/2)^{a/2}}{\sqrt{2\pi}\Gamma(a/2)} \frac{\Gamma((a+1)/2)}{(b/2+(x-m)^2/2)^{(a+1)/2}}$$
$$= \frac{\Gamma((a+1)/2)}{\sqrt{b/a}\sqrt{a\pi}\Gamma(a/2)} \left(1 - \frac{\left((x-m)/\sqrt{b/a}\right)^2}{a}\right)^{-(a+1)/2}$$

The above function is the p.d.f. of Student's t location-scale distribution with a degrees of freedom, center m and scale parameter $\sqrt{b/a}$.

2 The multivariate normal distribution

2.1 Basics

We all know the univariate normal distribution, whose long history began with de Moivre's 18th-century work on approximating the (analytically inconvenient) binomial distribution. This led to the probability density function

$$p(x) = \frac{1}{\sqrt{2\pi v}} \exp\left\{-\frac{(x-m)^2}{2v}\right\}$$

for the normal random variable with mean m and variance v, written $x \sim \mathcal{N}(m, v)$.

Here's an alternative characterization of the univariate normal distribution in terms of moment-generating functions:[4] a random variable x has a normal distribution if and only if $E\{\exp(tx)\} = \exp(mt + vt^2/2)$ for some real m and positive real v. Remember that $E(\cdot)$ denotes the expected value of its argument under the given probability distribution. We will generalize this definition to the multivariate normal.

(A) First, some simple moment identities. The covariance matrix $\text{cov}(x)$ of a vector-valued random variable x is defined as the matrix whose (i,j) entry is the covariance between x_i and x_j. In matrix notation, $\text{cov}(x) = E\{(x-\mu)(x-\mu)^T\}$, where μ is the mean vector whose ith component is $E(x_i)$. Prove the following: (1) $\text{cov}(x) = E(xx^T) - \mu\mu^T$; and (2) $\text{cov}(Ax+b) = A\text{cov}(x)A^T$ for matrix A and vector b.

(B) Consider the random vector $z = (z_1, \ldots, z_p)^T$, with each entry having an independent standard normal distribution (that is, mean 0 and variance 1). Derive the probability density function

[4] Laplace transforms to everybody but statisticians.

(PDF) and moment-generating function (MGF) of z, expressed in vector notation.[5] We say that z has a standard multivariate normal distribution.

(C) A vector-valued random variable $x = (x_1, \ldots, x_p)^T$ has a *multivariate normal distribution* if and only if every linear combination of its components is univariate normal. That is, for all vectors a not identically zero, the scalar quantity $z = a^T x$ is normally distributed. From this definition, prove that x is multivariate normal, written $x \sim \mathcal{N}(\mu, \Sigma)$, if and only if its moment-generating function is of the form $E(\exp\{t^T x\}) = \exp(t^T \mu + t^T \Sigma t/2)$. Hint: what are the mean, variance, and moment-generating function of z, expressed in terms of moments of x?

(D) Another basic theorem is that a random vector is multivariate normal if and only if it is an affine transformation of independent univariate normals. You will first prove the "if" statement. Let z have a standard multivariate normal distribution, and define the random vector $x = Lz + \mu$ for some $p \times p$ matrix L of full column rank.[6] Prove that x is multivariate normal. In addition, use the moment identities you proved above to compute the expected value and covariance matrix of x.

(E) Now for the "only if." Suppose that x has a multivariate normal distribution. Prove that x can be written as an affine transformation of standard normal random variables. (Note: a good way to prove that something can be done is to do it! Think about a matrix A such that $AA^T = \Sigma$.) Use this insight to propose an algorithm for simulating multivariate normal random variables with a specified mean and covariance matrix.

(F) Use this last result, together with the PDF of a standard multivariate normal, to show that the PDF of a multivariate normal $x \sim \mathcal{N}(\mu, \Sigma)$ takes the form $p(x) = C \exp\{-Q(x-\mu)/2\}$ for some constant C and quadratic form $Q(x - \mu)$.[7]

(G) Let $x_1 \sim N(\mu_1, \Sigma_1)$ and $x_2 \sim N(\mu_2, \Sigma_2)$, where x_1 and x_2 are independent of each other. Let $y = Ax_1 + Bx_2$ for matrices A, B of full column rank and appropriate dimension. Note that x_1 and x_2 need not have the same dimension, as long as Ax_1 and Bx_2 do. Use your previous results to characterize the distribution of y.

Answer

(A) (1)

$$\begin{aligned} cov(x) &= E((x-\mu)(x-\mu)^T) \\ &= E(xx^T) - E(x)\mu^T - \mu E(x^T) + \mu\mu^T \\ &= E(xx^T) - \mu\mu^T \end{aligned}$$

Here we use the fact that $E(x) = \mu$.

[5] Remember that the MGF of a vector-valued random variable x is the expected value of the quantity $\exp(t^T x)$, as a function of the vector argument t.

[6] The full rank restriction turns out to be unnecessary; relaxing it leads to what is called the *singular normal distribution*.

[7] A useful fact is that the Jacobian matrix of the linear map $x \to Ax$ is simply A.

(2)

$$\begin{aligned} cov(Ax+b) &= E(A(x-\mu)(x-\mu)A^T) \\ &= AE((x-\mu)(x-\mu)^T)A^T \\ &= Acov(x)A^T \end{aligned}$$

(B) The p.d.f. is:

$$p(z_1,\ldots,z_p) = \Pi_{i=1}^p \frac{1}{\sqrt{2\pi}} exp(-z_i^2/2) = \frac{1}{(2\pi)^{p/2}} exp(-z^T z/2)$$

The m.g.f. is:

$$\begin{aligned} M_z(t) \coloneqq E(exp(t^T z)) &= \int_{z\in\mathbb{R}^p} exp(t^T z)p(z)dz \\ &= \Pi_{i=1}^p \int_{z_i\in\mathbb{R}} exp(t_i z_i)p(z_i)dz_i \\ &= \Pi_{i=1}^p exp(mt_i + vt_i^2/2) \\ &= exp(m\mathbf{1}^T t + vt^T t/2) \end{aligned}$$

(C) (Sufficiency) If $M_x(t) = exp(t^T\mu + t^T\Sigma t/2)$, then $M_z(t) = M_x(ta) = exp\left((a^T\mu)t + (a^T\Sigma a))t^2/2\right)$. This is the m.g.f. of a normal distribution of mean $a^T\mu$ and variance $a^T\Sigma a$, so by arbitrariness of a, we know that every linear combination of x is univariate normal and thus x is multivariate normal. The mean of x is $\nabla M_x(\mathbf{0}) = (\mu + \Sigma t)M_x(t) \mid_{t=\mathbf{0}} = \mu$ and the variance is $\mathcal{H}_{M_x}(\mathbf{0}) = \left((\mu+\Sigma t)(\mu+\Sigma t)^T + \Sigma\right) M_x(t) \mid_{t=\mathbf{0}} = \Sigma$.

(Necessity) If x is multivariate normal, then for any $a \in \mathcal{R}^p$

$$M_x(ta) = M_{a^T x}(t) = exp\left((a^T\mu)^T t + t^T(a^T\Sigma a)t/2\right) = exp\left(\mu^T(ta) + (ta)^T\Sigma(ta)/2\right)$$

(D) (Sufficiency) The m.g.f. of x is:

$$M_x(t) = exp(t^T\mu)M_z(L^T t) = exp(t^T\mu + t^T(LL^T)t/2)$$

By (C), we know that x is multivariate normal.

(E) (Necessity) If $x \sim \mathcal{N}(\mu,\Sigma)$, then $z = \Sigma^{-1/2}(x-\mu)$ is a vector of independent univariate normals.

(F) By change of variable theorem,

$$\begin{aligned} p(x) &= p(z(x)) \left| det\left(\frac{\partial z}{\partial x}\right)\right| \\ &= \frac{1}{(2\pi)^{p/2}} exp\left(-\frac{1}{2}(\Sigma^{-1/2}(x-\mu))^T(\Sigma^{-1/2}(x-\mu))\right) |\Sigma|^{-1/2} \\ &= \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \end{aligned}$$

(G) The m.g.f. of y is:

$$M_y(t) = M_{x_1}(A^T t) M_{x_2}(B^T t) = exp\left(t^T(A\mu_1 + B\mu_2) + t^T(A\Sigma_1 A^T + B\Sigma_2 B^T)t/2\right)$$

There, $y \sim \mathcal{N}(A\mu_1 + B\mu_2, A\Sigma_1 A^T + B\Sigma_2 B^T)$

2.2 Conditionals and marginals

Suppose that $x \sim \mathcal{N}(\mu, \Sigma)$ has a multivariate normal distribution. Let x_1 and x_2 denote an arbitrary partition of x into two sets of components. Because we can relabel the components of x without changing their distribution, we can safely assume that x_1 comprises the first k elements of x, and x_2 the last $p - k$. We will also assume that μ and Σ have been partitioned conformably with x:

$$\mu = (\mu_1, \mu_2)^T \quad \text{and} \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}.$$

Clearly $\Sigma_{21} = \Sigma_{12}^T$, as Σ is a symmetric matrix.

(A) Derive the marginal distribution of x_1. (Remember your result about affine transformations.)

(B) Let $\Omega = \Sigma^{-1}$ be the inverse covariance matrix, or precision matrix, of x, and partition Ω just as you did Σ:

$$\Omega = \begin{pmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{12}^T & \Omega_{22} \end{pmatrix}.$$

Using (or deriving!) identities for the inverse of a partitioned matrix, express each block of Ω in terms of blocks of Σ.

(C) Derive the conditional distribution for x_1, given x_2, in terms of the partitioned elements of x, μ, and Σ. There are several keys to inner peace: work with densities on a log scale, ignore constants that don't affect x_1, and remember the cute trick of completing the square from basic algebra.[8] Explain briefly how one may interpret this conditional distribution as a linear regression on x_2, where the regression matrix can be read off the precision matrix.

Answer

(A) Let $A = \begin{pmatrix} \mathbf{I}_{k,k} & \mathbf{O}_{k,p-k} \end{pmatrix}$, then $x_1 = Ax$. According to 2.1(G), we know $x_1 \sim \mathcal{N}(\mu_1, \Sigma_{11})$

(B) Notice,

$$\begin{pmatrix} \Delta_{11}^{-1} & O \\ 0 & \Sigma_{22}^{-1} \end{pmatrix} \begin{pmatrix} I & O \\ -\Sigma_{21}\Delta_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} I & -\Sigma_{12}\Sigma_{22}^{-1} \\ O & I \end{pmatrix} \Sigma = I$$

where $\Delta_{11} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$.

[8] In scalar form:

$$\begin{aligned} x^2 - 2bx + c &= x^2 - 2bx + b^2 - b^2 + c \\ &= (x - b)^2 - b^2 + c\,. \end{aligned}$$

Therefore, $\Omega = \begin{bmatrix} \Delta_{11}^{-1} & -\Delta_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1} \\ -\Sigma_{22}^{-1}\Sigma_{21}\Delta_{11}^{-1} & \Sigma_{22}^{-1} - \Sigma_{22}^{-1}\Sigma_{21}\Delta_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1} \end{bmatrix}$.

(C) Notice, $y_1 = x_1 - \Sigma_{12}\Sigma_{22}^{-1}x_2$ is independent of x_2. It is because $cov(y_1, x_2) = 0$ and the joint distribution of (y_1, x_2) is multivariate normal. Therefore, $x_1|x_2 = y_1 + \Sigma_{12}\Sigma_{22}^{-1}x_2|x_2 \sim \mathcal{N}(E(y_1)+\Sigma_{12}\Sigma_{22}^{-1}x_2, cov(y_1))$, where $E(y_1) = \mu_1 - \Sigma_{12}\Sigma_{22}^{-1}\mu_2$ and $cov(y_1) = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$

3 Multiple regression: three classical principles for inference

Suppose we observe data that we believe to follow a linear model, where $y_i = x_i^T\beta + \epsilon_i$ for $i = 1, \ldots, n$. To fix notation: y_i is a scalar response; x_i is a p-vector of predictors or features; and the ϵ_i are errors. By convention we write vectors as column vectors. Thus $x_i^T\beta$ will be our typical way of writing the inner product between the vectors x_i and β.pc[-4]

Consider three classic inferential principles that are widely used to estimate β, the vector of regression coefficients. In this context we will let $\hat{\beta}$ denote an estimate of β, $y = (y_1, \ldots, y_n)^T$ the vector of outcomes, X the matrix of predictors whose ith row is x_i^T, and ϵ the vector of residuals $(\epsilon_1, \ldots, \epsilon_n)^T$.

Least squares: make the sum of squared errors as small as possible. We can express this in terms of the squared Euclidean norm of the residual vector $\epsilon = y - X\beta$:

$$\hat{\beta} = \arg\min_{\beta \in \mathcal{R}^p} \|y - X\beta\|_2^2 = \arg\min_{\beta \in \mathcal{R}^p} (y - X\beta)^T(y - X\beta)$$

Maximum likelihood under Gaussianity: assume that the errors are independent, mean-zero normal random variables with common variance σ^2. Choose $\hat{\beta}$ to maximize the likelihood:

$$\hat{\beta} = \arg\max_{\beta \in \mathcal{R}^p} \left\{ \prod_{i=1}^{n} p(y_i \mid \beta, \sigma^2) \right\}.$$

Here $p_i(y_i \mid \sigma^2)$ is the conditional probability density function of y_i, given the model parameters β and σ^2. Note that an equivalent way to write the likelihood is to say that the response vector y is multivariate normal with mean $X\beta$ and covariance matrix $\sigma^2 I$, where I is the n-dimensional identity matrix.

Method of moments: Choose $\hat{\beta}$ so that the sample covariance between the errors and each of the p predictors is exactly zero. (That is, the sample covariance of ϵ and each column of X is zero.) This gives you a system of p equations and p unknowns.

(A) Show that all three of these principles lead to the same estimator. What is the variance of this estimator under the assumption that each ϵ_i is independent and identically distribution with variance σ^2?

[-4]Notice we have no explicit intercept. For now you can imagine that all the variables have had their sample means subtracted, making an intercept superfluous. Or you can just assume that the leading entry in every x_i is equal to 1, in which case β_1 will be an intercept term.

(B) As mentioned above, the estimator in the previous part corresponds to the assumption that $y \sim N(X\beta, \sigma^2 I)$. What happens if we instead postulate that $y \sim N(X\beta, \Sigma)$, where Σ is an arbitrary known covariance matrix, not necessarily proportional to the identity? What is the maximum likelihood estimate for β now, and what is the variance of this estimator?

(C) Show that in the special case where Σ is a diagonal matrix, i.e. $\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2)$, that the MLE is the familiar *weighted least squares* estimator. That is, show that $\hat{\beta}$ is the solution to the following linear system of P equations in P unknowns:

$$(X^T W X)\hat{\beta} = X^T W y\,,$$

where W is a diagonal matrix of weights that you should relate to the σ_i^2's.

Answer

(A) (LS) Assume $X^T X \in \mathbb{R}^{p\times p}$ is invertible, then it is positive definite, the problem is a typical strictly convex optimization, where there is a unique solution $\hat{\beta}$ that satisfies first order condition

$$X^T(y - X\hat{\beta}) = 0 \iff \hat{\beta} = (X^T X)^{-1} X^T y$$

(MLE) The log likelihood is:

$$\begin{aligned} l(\beta) = log(\mathbf{y}|X, \beta, \sigma^2) &= \sum_{i=1}^{n} logp(y_i|x_i, \beta, \sigma^2) \\ &\propto -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 \\ &= -\frac{1}{2\sigma^2}(\mathbf{y} - X\beta)^T(\mathbf{y} - X\beta) \end{aligned}$$

To maximize the likelihood is equivalent to minimize the log-likelihood. Thus, MLE has the same objective and solution as least squares.

(MoM) The independence between the noise and predictors gives us an equation

$$\begin{aligned} \widehat{cov}(\hat{\epsilon}, x) = 0 &\iff \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^T\beta)x_i = 0 \\ &\iff (y - X\beta)^T X = 0 \end{aligned}$$

which is exactly the same equation we obtained earlier by first order condition for the least squares problem.

(B) Now the likelihood becomes:

$$log(\mathbf{y}|X, \beta, \sigma^2) = -\frac{1}{2}(\mathbf{y} - X\beta)^T\Sigma^{-1}(\mathbf{y} - X\beta)$$

Similarly, by first order condition, we have

$$X^T\Sigma^{-1}(y - X\hat{\beta}) = 0 \iff \hat{\beta} = (X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}y$$

The covariance of this estimator is

$$\begin{aligned} cov(\hat{beta}) &= (X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1})cov(y)((X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1})^T \\ &= (X^T\Sigma^{-1}X)^{-1} \end{aligned}$$

(C) If Σ is diagonal, then it becomes trivial to calculate the inverse of Σ. In fact, we have

$$\Sigma^{-1} = diag(1/\sigma_1^2, 1/\sigma_2^2, \ldots, 1/\sigma_n^2)$$

Plug the equation into the formula we obtained in (B), we can immediately show that $W = \Sigma^{-1}$ is diagonal and each diagonal element is the inverse of σ_i^2's.

4 Some practical details

(A) Let's continue with the weighted least-squares estimator you just characterized, i.e. the solution to the linear system

$$(X^TWX)\hat{\beta} = X^TWy\,,$$

One way to calculate $\hat{\beta}$ is to: (1) recognize that, trivially, the solution to the above linear system must satisfy $\hat{\beta} = (X^TWX)^{-1}X^TWy$; and (2) to calculate this directly, i.e. by inverting X^TWX. Let's call this the "inversion method" for calculating the WLS solution.

Numerically speaking, is the inversion method the fastest and most stable way to actually solve the above linear system? Do some independent sleuthing on this question.[9]. Summarize what you find, and provide pseudo-code for at least one alternate method based on matrix factorizations—call it "your method" for short.[10]

(B) Code up functions that implement both the inversion method and your method for an arbitrary X, y, and set of weights W. Obviously you shouldn't write your own linear algebra routines for doing things like multiplying or decomposing matrices. But don't use a direct model-fitting function like R's "lm" either. Your actual code should look a lot like the pseudo-code you wrote for the previous part.[11]

Now simulate some silly data from the linear model for a range of values of N and P. (Feel free to assume that the weights w_i are all 1.) It doesn't matter how you do this—e.g. everything can be Gaussian if you want. (We're not concerned with statistical principles in this problem, just with algorithms, and using least squares is a pretty terrible idea for enormous linear models, anyway.) Just make sure that you explore values of P up into the thousands, and that $N > P$. Benchmark the performance of the inversion solver and your solver across a range of scenarios.[12]

[9] `https://www.google.com/search?q=Why+Shouldn\%27t+I+Invert+That+Matrix`

[10] Our linear system is not a special flower; whatever you discover about general linear systems should apply here.

[11] Be attentive to how you multiply a matrix by a diagonal matrix, or you'll waste a lot of time multiplying stuff by zero.

[12] In R, a simple library for this purpose is microbenchmark.

Answer

(A) No, the inversion method is neither the fastest nor the most stable way to actually solve the linear system. Firstly, the computational complexity of inverting a matrix is always going to outweigh simply performing corresponding LU decomposition. Secondly, for nearly singular matrices, computing the inverse can cause troubles due to numerical instability (e.g. machine precision restriction and rounding errors). One approach is to use LU decomposition:

1 Let A denote X^TWX, b denote X^TWy and x denote $\hat{\beta}$;

2 Perform PLU decomposition on A, i.e. $A = PLU$;

3 Permute b with P^{-1};

4 Forward-solve $Ly = b$;

5 Backward-solve $Ux = y$.

(B)
```
library(Matrix)
library(microbenchmark)
library(knitr)


lm_inv <- function(X,y,weights){
  W = diag(weights)
  A = t(X)%*%W%*%X
  b = t(X)%*%W%*%y
  A_inv = solve(A)
  return(A_inv%*%b)
}

lm_LU <- function(X, y, weights){
  W = diag(weights)
  A = t(X)%*%W%*%X
  p = nrow(A)
  b = t(X)%*%W%*%y
  eluA = expand(lu(A))
  b = b[sort(apply(eluA$P,1,function(x){which(x!=0)}),index.return=T)$ix]
  b = forwardsolve(eluA$L,b)
  b = backsolve(eluA$U,b)
  return(b)
}

lm_chol <- function(X, y, weights){
  W = diag(weights)
  A = t(X)%*%W%*%X
  p = nrow(A)
  b = t(X)%*%W%*%y
  U = base::chol(A)
  b = forwardsolve(t(U),b)
```

```
  b = backsolve(U,b)
  return(b)
}

bm_setup <- function(n, p){
  beta = rnorm(p)
  sigma = rexp(n)
  X = matrix(rnorm(n*(p-1)), nrow=n)
  X = cbind(rep(1,n), X)
  y = X%*%beta+sigma*rnorm(n)
  weights = 1/sigma^2
}

ps = c(50,100,500,1000)
bm_inv = list()
for(i in 1:4){
  p = ps[i]
  n = 2*p
  cat(sprintf("p=%d:\n",p))
  bm_inv[[i]] = microbenchmark(lm_inv(X,y,weights),times=10,setup={
    beta = rnorm(p)
    sigma = rexp(n)
    X = matrix(rnorm(n*(p-1)), nrow=n)
    X = cbind(rep(1,n), X)
    y = X%*%beta+sigma*rnorm(n)
    weights = 1/sigma^2
  })
}
df_inv = do.call(rbind,lapply(bm_inv,function(x){summary(x,unit="ms")[,c(-1,-8)]}))
df_inv = data.frame(p=ps, df_inv)
tab_inv = kable(df_inv, "latex", booktabs=T)

bm_lu = list()
for(i in 1:4){
  p = ps[i]
  n = 2*p
  cat(sprintf("p=%d:\n",p))
  bm_lu[[i]] = microbenchmark(lm_LU(X,y,weights),times=10,setup={
    beta = rnorm(p)
    sigma = rexp(n)
    X = matrix(rnorm(n*(p-1)), nrow=n)
    X = cbind(rep(1,n), X)
    y = X%*%beta+sigma*rnorm(n)
    weights = 1/sigma^2
  })
}
```

```
df_lu = do.call(rbind,lapply(bm_lu,function(x){summary(x,unit="ms")[,c(-1,-8)]}))
df_lu = data.frame(p=ps, df_lu)
tab_lu = kable(df_lu, "latex", booktabs=T)

bm_chol = list()
for(i in 1:4){
  p = ps[i]
  n = 2*p
  cat(sprintf("p=%d:\n",p))
  bm_chol[[i]] = microbenchmark(lm_inv(X,y,weights),times=10,setup={
    beta = rnorm(p)
    sigma = rexp(n)
    X = matrix(rnorm(n*(p-1)), nrow=n)
    X = cbind(rep(1,n), X)
    y = X%*%beta+sigma*rnorm(n)
    weights = 1/sigma^2
  })
}
df_chol = do.call(rbind,lapply(bm_chol,function(x){summary(x,unit="ms")[,c(-1,-8)]}))
df_chol = data.frame(p=ps, df_chol)
tab_chol = kable(df_chol, "latex", booktabs=T)
```

p	min	lq	mean	median	uq	max
50	0.9802	1.0493	1.27174	1.21810	1.3018	1.7086
100	7.0462	7.1249	7.48689	7.55985	7.7408	8.0715
500	796.4476	828.9354	859.97342	847.05225	898.6678	939.3487
1000	8078.7876	8137.0667	8194.85307	8187.78965	8248.0179	8326.6004

Table 1: Benchmark results of naive inverse solver

p	min	lq	mean	median	uq	max
50	2.2539	2.3671	3.54891	2.62520	3.0491	11.7697
100	8.6220	8.9029	10.02539	9.84945	11.3215	12.3723
500	819.3632	824.6924	943.87182	908.25625	1039.6591	1159.7416
1000	7469.4160	7476.4963	7541.93532	7493.86930	7541.8868	7907.8093

Table 2: Benchmark results of LU decomposition solver