

1 Contours and Eigenvalues of the covariance matrix

Let's look at the density of the d-dimensional multivariate Gaussian again.

$$p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp(-(x - \mu)^T \Sigma^{-1} (x - \mu)/2)$$

1.1 Diagonal covariance matrix

Let's first consider a simple 2-d case where

$$\mu = (0, 0)^T$$

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$$

$$\Sigma^{-1} = \begin{pmatrix} 1/\sigma_1^2 & 0 \\ 0 & 1/\sigma_2^2 \end{pmatrix}$$

We get

$$\begin{aligned} p(x|\mu, \Sigma) &= \frac{1}{2\pi\sigma_1\sigma_2} \exp(-(x_1, x_2)^T \Sigma^{-1} (x_1, x_2)/2) \\ &= \frac{1}{2\pi\sigma_1\sigma_2} \exp(-x_1 \frac{1}{\sigma_1^2} x_1/2 - x_2 \frac{1}{\sigma_2^2} x_2/2) \\ &= \frac{1}{\sqrt{2\pi}\sigma_1} \exp(-x_1^2/2\sigma_1^2) \frac{1}{\sqrt{2\pi}\sigma_2} \exp(-x_2^2/2\sigma_2^2) \\ &= p(x_1|\mu_1, \sigma_1^2) p(x_2|\mu_2, \sigma_2^2) \end{aligned}$$

This is just the density for 2 independent normals.

What does this look like in terms of the contours of the density.

$$\begin{aligned} \log p(x|\mu, \Sigma) &= \log p(x_1|\mu_1, \sigma_1^2) + \log p(x_2|\mu_2, \sigma_2^2) \\ &= \text{const} - \frac{x_1^2}{2\sigma_1^2} - \frac{x_2^2}{2\sigma_2^2} \end{aligned}$$

Thus each contour consist of the points that satisfy an equation of the form

$$c^2 = \frac{x_1^2}{\sigma_1^2} + \frac{x_2^2}{\sigma_2^2}$$

Draw the contour plots. Axes are the major/minor of the ellipse.

1.2 General covariance matrix

Now consider a general (positive definite) covariance matrix Σ . We keep $\mu = 0$ We can use the spectral theorem to decompose the matrix

$$\Sigma = U^T \Lambda U$$

where U is an orthogonal matrix and Λ is a diagonal matrix where the diagonal entries are > 0

We also get the inverse

$$\Sigma^{-1} = (U^T \Lambda U)^{-1} = U^{-1} \Lambda^{-1} U^{-T} = U^T \Lambda^{-1} U$$

since U orthogonal implies $U^{-1} = U^T$

The diagonal entries of Λ are the eigenvalues and the corresponding rows of U are the corresponding eigenvectors.

$$\begin{aligned} \log p(x|\mu, \Sigma) &= \text{const} - x^T \Sigma^{-1} x / 2 \\ &= \text{const} - x^T U^T \Lambda^{-1} U x / 2 \\ &= \text{const} - (Ux)^T \Lambda^{-1} (Ux) / 2 \\ &= \text{const} - \sum_i \frac{x_i'^2}{\lambda_i^2} \end{aligned}$$

Reparameterizing $x' = Ux$ brings us back to the diagonal case.

Graphically what does this mean? Orthogonal matrices can be viewed as generalized rotation matrices.

Work out a 2d example.

$$\Sigma = \begin{pmatrix} \cos(30) & \sin(30) \\ -\cos(30) & \cos(30) \end{pmatrix}^T \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}^T \begin{pmatrix} \cos(30) & \sin(30) \\ -\cos(30) & \cos(30) \end{pmatrix}$$

We get the major axis along $(\cos(30), \sin(30))$ and minor axis along $(\cos(30), \sin(30))$.

What if we take $\mu \neq 0$? Then we do the same kind of reparameterization trick $x' = U(x - \mu)$. Geometrically this just means we shift the center of the ellipse

We can also flip this around. Given an $n \times n$ matrix A and a vector x of n independent standard normals, Ax is $Normal(0, AA^T)$ eg we may take $A = U^T \Lambda^{1/2}$

1.3 Distance metric viewpoint

We may also view the map $d(x, y)^2 = (x - y)^T \Sigma^{-1} (x - y)$ as defining a distance metric and $(x, y) \rightarrow x^T \Sigma^{-1} y$ as defining an inner product.

In Euclidean space, we have

$$\begin{aligned}\langle x, y \rangle &= x^T y \\ \|x\|^2 &= x^T x\end{aligned}$$

It is easy to verify all the properties of inner product are satisfied by

$$\langle x, y \rangle_{\Sigma^{-1}} = x^T \Sigma^{-1} y$$

Thus, the following is a distance metric

$$d(x, y)_{\Sigma^{-1}}^2 = (x - y)^T \Sigma^{-1} (x - y)$$

What does this mean geometrically? Go back to the spectral decomposition $\Sigma = U^T \Lambda U$. To go from our “new” space back down to usual Euclidean space, we stretch it along the eigenvectors. The amount of stretching along eigenvector v_i is equal to the square root of the corresponding eigenvalue $\sqrt{\lambda_i}$.

What does this mean in terms of normal densities? The log of the density is given by

$$\log p(x|\mu, \Sigma) = \text{const} - d(x, \mu)_{\Sigma^{-1}}^2/2$$

The contours (ie the curves of equal density) are exactly the points equidistant from the mean.

Example: classification

Consider two multivariate normals with known mean and covariance. Assume the covariance matrix is the same, but the means are different.

Given a data point, which normal did it come from?

Setting the posterior log odds ratio to 0 gets us the decision boundary

$$\begin{aligned}\log \frac{p(Y=0|X)}{p(Y=1|X)} &= \log \frac{p(X|Y=0) p(Y=0)}{p(X|Y=1) p(Y=1)} \\ &= \log p(X|Y=0) - \log p(X|Y=1) + \log \frac{p(Y=0)}{p(Y=1)} \\ &= -d(X, \mu_0)_{\Sigma^{-1}}^2/2 + d(X, \mu_1)_{\Sigma^{-1}}^2/2 + \log \frac{p(Y=0)}{p(Y=1)}\end{aligned}$$

Stop here and draw contour plots

$$d(X, \mu_0)_{\Sigma^{-1}}^2/2 = d(X, \mu_1)_{\Sigma^{-1}}^2/2 + \log \frac{p(Y=0)}{p(Y=1)}$$

Note that if Σ is not a multiple of the identity then the lines (in Euclidean space) are not orthogonal to the line between the means.

Continuing to get an algebraic solution

$$\begin{aligned}\log \frac{p(Y=0|X)}{p(Y=1|X)} &= -\frac{1}{2}(X - \mu_0)^T \Sigma^{-1} (X - \mu_0) + \frac{1}{2}(X - \mu_1)^T \Sigma^{-1} (X - \mu_1) + \log \frac{p(Y=0)}{p(Y=1)} \\ &= -\mu_0^T \Sigma^{-1} X + \mu_1^T \Sigma^{-1} X - \mu_0^T \Sigma^{-1} \mu_0 + \mu_1^T \Sigma^{-1} \mu_1 + \log \frac{p(Y=0)}{p(Y=1)} \\ &= (\mu_1 - \mu_0)^T \Sigma^{-1} X - \mu_0^T \Sigma^{-1} \mu_0 + \mu_1^T \Sigma^{-1} \mu_1 + \log \frac{p(Y=0)}{p(Y=1)} \\ &= \beta^T X + \gamma\end{aligned}$$

When the covariance matrices are different Σ_1, Σ_2 define 2 distance metrics. We can't "reshape" the entire space by applying a single distance metric.

Draw some rough contour plots showing the quadratic relationship.

Algebraically, this happens since you are left with a quadratic term

$$x^T \Sigma_1^{-1} x - x^T \Sigma_2^{-1} x$$

2 Generalized Linear Models (GLIMs) and the Iteratively Reweighted Least Squares (IRLS) algo

We mainly want to go over IRLS, but you'll see more about the GLIM model itself in class tomorrow.

In the usual regression setting we had the model

$$E[Y|x] = \beta^T x$$

Least squares linear regression gave the ML solution when $Y \sim \text{Normal}(\beta^T x, \sigma^2)$

We can generalize linear regression by considering the following model

$$E[Y|x] = \mu = f(\theta^T x)$$

In this model, we assume

- x enters model via linear combination $\theta^T x$
- μ is a function of this linear combination, $f(\theta^T x)$.
- y is distributed according to an exponential family function, with conditional mean $\mu = f(\theta^T x)$.

This model is a Generalized Linear Model (GLIM). The function f is called the *response function*. f^{-1} is called the *link function*

The GLIM framework extends linear regression. It

- is useful for modeling different types of data (binary, counts, etc)
- lends itself to an easy algorithm (IRLS) for finding MLE for the parameters θ .

Ex: Logistic regression is an example of this where the response and link functions are

$$f(x) = \frac{1}{1 + e^{-x}}$$

$$f^{-1}(y) = \log \frac{y}{1 - y}$$

This is often used when the dependent variable Y takes binary values (0-1). $E[Y|x]$ gives the Bernoulli probability of success. ie $Y|x \sim \text{Bernoulli}(f(\theta^T x))$

The response function scales the values into the interval (0, 1)

When we model data Y according to a GLIM, we need to choose:

- Which exponential family
- Which response function

Which exponential family we choose usually depends on the type of data. eg logistic for binary, poisson for counts.

Once an exponential family is chosen, we often choose the *canonical response function*. Recall that for exponential families we have the natural parameterization η and the moment parameterization μ . There is an invertible mapping $\eta = \psi(\mu)$ between the 2 parameterizations.

The *canonical response function* is the function

$$f(\theta^T x) = f(\eta) = \phi^{-1}(\eta) = A'(\eta) = E[Y|\eta] = E[Y|x]$$

We may now ask, what value of the parameters θ gives the MLE?

3 Newton-Raphson

Newton-Raphson is a general, iterative algorithm to maximize (or minimize) a scalar valued function $J(\theta)$.

The general form of one iteration is

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - H^{-1} \nabla_{\theta} J$$

where $\nabla_{\theta} J$ is the gradient vector of $J(\theta)$ and H is the Hessian matrix of $J(\theta)$ (the second derivative of J with respect to θ).

3.1 1-dim case

It is easiest to gain an intuition of Newton-Raphson in the 1-dimensional case.

Consider a concave quadratic polynomial

$$f(x) = a(x - x_0)^2 + b(x - x_0) + c$$

by doing a Taylor expansion.

The gradient and Hessian are

$$f'(x) = 2a(x - x_0) + b$$

$$f''(x) = 2a$$

To maximize f we set the gradient to 0

$$0 = 2a(x_{min} - x_0) + b$$

$$x_{min} = x_0 - \frac{b}{2a} = x_0 - \frac{f'(x_0)}{f''(x_0)}$$

This is exactly the Newton-Raphson step, so for a convex quadratic polynomial, it jumps to the maximum in 1-step.

Consider an arbitrary convex function on 1 variable, and carry out a Taylor series expansion

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + f''(x_0)(x - x_0)^2$$

The r.h.s. is just like the quadratic polynomial case. We can maximize it just like before. We hope that by maximizing an approximation to f , we can get closer to the true maximizer.

This gives us intuition about how Newton-Raphson works.

At each step, Newton-Raphson approximates the function of interest with a quadratic approximation and finds the maximizer of the approximation.

(Draw a picture)

3.2 Newton-Raphson as choosing a step size

Return to the 1-dim form of Newton-Raphson.

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - J'(\theta)/J''(\theta)$$

This looks like a gradient method with step size chosen to be

$$\rho = \frac{1}{J''(\theta)}$$

As the iterates get closer to the solution, the step size gets smaller.

3.3 Multivariate Quadratic functions

Ex: $J(\theta)$ is the negative sum of squared error (written out in matrix notation):

$$J(\theta) = -\frac{1}{2}(y - X\theta)^T(y - X\theta)$$

then:

$$\nabla_{\theta} J = X^T(y - X\theta)$$

and

$$H = -X^T X.$$

Then the general form of a Newton Raphson batch algorithm is:

$$\theta^{(t+1)} \leftarrow \theta^{(t)} + (X^T X)^{-1} X^T (y - X\theta^t)$$

or

$$\theta^{(t+1)} \leftarrow (X^T X)^{-1} X^T y$$

Hops to the final solution in a single step!!

The same intuition that a Newton-Raphson iteration maximizes a quadratic approximation to the true function still holds in the multivariate case.

4 IRLS

In general, we can take J to be the log likelihood function $l(\theta)$. Finding the maximizer of J then means finding the MLE.

Recall:

$$l(\theta|\mathcal{D}) = \log \prod_{n=1}^N h(y_n) \exp\{\eta_n^T y_n - A(\eta_n)\}$$

where $\eta_n = \psi(\mu_n)$ and $\mu_n = f(\theta^T x_n)$, and we are choosing f to be the canonical response function, so $\eta_n = \theta^T x_n$:

$$\begin{aligned} &= \sum_{n=1}^N \log h(y_n) + \sum_{n=1}^N (\theta^T x_n y_n - A(\eta_n)) \\ &= \sum_{n=1}^N \log h(y_n) + \theta^T \sum_{n=1}^N x_n y_n - \sum_{n=1}^N A(\eta_n) \end{aligned}$$

Now, take the gradient of the log likelihood with respect to θ (change of variables, since $\eta_n = \theta^T x_n$):

$$\begin{aligned} \nabla_{\theta} l &= \sum_{n=1}^N x_n y_n - A'(\eta_n) \frac{d\eta_n}{d\theta} \\ &= \sum_{n=1}^N x_n y_n - A'(\eta_n) x_n \\ &= \sum_{n=1}^N (y_n - \mu_n) x_n \end{aligned}$$

We can use this in the Newton-Raphson formulation, in vector notation:

$$\begin{aligned} \nabla_{\theta} l &= \sum_{n=1}^N (y_n - \mu_n) x_n \\ &= X^T (y - \mu) \end{aligned}$$

We also need the Hessian. Take another derivative:

$$H = - \sum_n \frac{d\mu_n}{d\eta_n} x_n x_n^T$$

or

$$H = X^T W X$$

where

$$W = \text{diag} \left\{ \frac{d\mu_1}{d\eta_1}, \dots, \frac{d\mu_N}{d\eta_N} \right\}$$

These are just the second derivatives of $A(\eta_n)$ (i.e., variance of Y_n).

Note that weight matrix depends on $\theta^{(t)}$ (superscript not included though).

Plugging this into the Newton-Raphson form:

$$\begin{aligned} \theta^{(t+1)} &\leftarrow \theta^{(t)} - H^{-1} \nabla_{\theta} J \\ &= (X^T W^{(t)} X)^{-1} \left[X^T W^{(t)} X \theta^{(t)} + X^T (y - \mu^{(t)}) \right] \end{aligned}$$

This is similar to the solution of weighted linear least squares regression

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

Why is that? Newton-Raphson uses a quadratic function to approximate the log-likelihood. The log-likelihood of a normal is a quadratic function.

Example: Logistic regression. Compute Hessian:

$$\nabla_{\eta_n} A(\eta_n) = \mu_n = E[y_n | x_n, \theta] = p(y_n = 1 | x_n, \theta) = \frac{1}{1 + \exp^{\theta^T x_n}}$$

so, since derivative of logistic function μ_n is $\mu_n(1 - \mu_n)$:

$$W = \text{diag} \{ \mu_1(1 - \mu_1), \dots, \mu_N(1 - \mu_N) \}$$

Leave it as an exercise for the reader to handle noncanonical response functions: idea is that Hessian matrix has an additional term. Alternatively, we can compute the expected Hessian.

4.1 Fisher's scoring method

For non-canonical link functions, the Hessian become uglier to compute. This is because

$$\eta_n = \psi(\mu_n) = \psi(f(\theta^T x_n)) \neq \theta^T x_n$$

Digression into multivariate calc.

In vector/matrix calculus notation, the gradient is

$$\nabla_{\theta} f(\theta) = \left(\frac{d}{d\theta} f(\theta) \right)^T =: \frac{d}{d\theta^T} f(\theta)$$

$\nabla_{\theta}|_{\theta_0} f(\theta)$ is a vector.

$\frac{d}{d\theta}|_{\theta_0} f(\theta)$ is a linear functional represented by the transpose of a vector.

$\nabla_{\theta} f(\theta)$ gives a vector valued function of θ .

The Hessian is the matrix of second order partial derivatives and equal to the derivative of this vector valued function.

$$H = \frac{d}{d\theta} \nabla_{\theta} f(\theta) = \frac{d^2}{d\theta\theta^T} f(\theta)$$

Put $\xi_n = \theta^T x_n$.

$$l(\theta|\mathcal{D}) = \text{const} + \sum_{n=1}^N (\eta_n y_n - A(\eta_n))$$

$$\nabla_{\theta} l = \sum_{n=1}^N \nabla_{\theta} \eta_n (y_n - A'(\eta_n))$$

Taking the Hessian then gives

$$\begin{aligned} H = \frac{d}{d\theta\theta^T} l &= \sum_{n=1}^N \frac{d^2 \eta_n}{d\theta\theta^T} (y_n - \mu_n) - \frac{d\eta_n}{d\theta^T} \frac{d\mu_n}{d\theta} \\ &= \sum_{n=1}^N \frac{d^2 \eta_n}{d\theta\theta^T} (y_n - \mu_n) - \left(\frac{d\eta_n}{d\xi_n} \frac{d\xi_n}{d\theta^T} \right) \left(\frac{d\mu_n}{d\eta_n} \frac{d\eta_n}{d\xi_n} \frac{d\xi_n}{d\theta} \right) \\ &= \frac{d^2 \eta_n}{d\theta\theta^T} \sum_{n=1}^N (y_n - \mu_n) - \left(\frac{d\eta_n}{d\xi_n} \right)^2 A''(\eta_n) x_n x_n^T \end{aligned}$$

With a canonical link function $\frac{d\eta_n}{d\theta^T} = x_n$ had a simple form. We don't have that anymore. In particular, $\frac{d\eta_n}{d\theta\theta^T}$ can be ugly to compute.

Fisher's idea is this. Instead of calculating the true Hessian, replace it with the *expected* Hessian. Since $EY_n = \mu_n$ the first term disappears. That leaves the second term in the sum

Some intuition: Ideally y_n should be centered on μ_n if our parameters are good. From the step size viewpoint, as long as our step sizes are going to 0 at an appropriate rate, we will still reach the maximizer.