Supplementary notes

Likelihood Probability Models and Point Estimates

1 Introduction

Many estimation problems may be formulated as looking for a parameter combination $\theta = (\theta_1, \cdots, \theta_d)$ that best explains the data. You need some measure of *goodness of fit* to quantify how well a model fits the data. *Likelihood* based goodness of fit measures are based on stochastic models of the data. That is to say, they are based on hypotheses involving some randomness, "stories", to explain how X (the dataset) was generated (examples below). This model involves the parameters, θ. The probability model, implicitly or explicitly, defines a probability distribution for X. This probability distribution depends on the parameters θ.

There are minor variations on the likelihood idea, depending on whether the items in the dataset are (modeled as) discrete or continuous random variables. We start with the case in which X is a collection of n real numbers (continuous random variables), which we express as $X = (X_1, \cdots, X_n) \in \mathbb{R}^n$. We suppose there is a probability density (PDF) for X that depends on the parameter set θ. We write this as $L(x, \theta)$. We write $X \sim L(\cdot, \theta)$ to say that X has $L(\cdot, \theta)$ as its PDF. As we said before, if X is fixed (the data are known) and we look at L as a function of θ, then L is called a *likelihood* function. This emphasizes the idea that L is related to probability but θ is not a random variable.

A *point estimate* of the parameter set is a function $\widehat{\theta}(x)$ that represents a guess at θ that is inferred from dataset x. As we will see later, the *Bayesian* philosophy of statistics replaces the point estimate $\widehat{\theta}$ with a *posterior* probability density $p(\theta|x)$ that represents our information and uncertainty about θ from the data x. For now, we discuss just point estimates and leave *uncertainty quantification* for later.

In the *frequentist* version of statistics, there are some true parameters $(\theta_{*,1}, \cdots, \theta_{*,n}) = \theta_*$. The dataset is one sample of the probability density $L(\cdot, \theta)$. The point estimate is $\widehat{\theta}(X)$. The point estimate is a random variable because it is a function of X, which is random. Theoretical frequentist statistics asks questions about this random variable, such as: what is the expected value of $\widehat{\theta}(X)$:

$$\mathrm{E}\Big[\,\widehat{\theta}(X)\,\Big] = \int \widehat{\theta}(x)\, L(x, \theta_*)\, dx \ .$$

Note that this expected value depends on the true value θ_*. The frequentist point of view is that if you could repeat the same "experiment" with the same "system" (the same θ_*), you get different datasets $X_1 \neq X_2 \neq X_3 \neq \cdots$ that are different samples of the distribution represented by $L(\cdot, \theta_*)$. These would lead to different point estimates $\theta_1 = \widehat{\theta}(X_1)$, $\theta_2 = \widehat{\theta}(X_2), \cdots$. In reality, it is probably not possible to repeat experiments. There would be a single dataset X and a single corresponding $\widehat{\theta}$.

Maximum likelihood estimation uses the likelihood function as a measure of goodness of fit. The intuition is: higher the probability of getting X from parameter set θ, the greater the "likelihood" that θ is the true parameter set. The maximum likelihood estimate (often called *MLE*) the θ that maximizes this likelihood:

$$\widehat{\theta}_{\rm ML}(X) = \arg\ \max_{\theta} L(X|\theta) \ . \tag{1}$$

The reasoning behind maximum likelihood estimation is not 'rigorous" and the MLE is not necessarily the recommended estimator even when there is a natural likelihood function.

There are practical issues that can make the optimization problem (1) hard to solve. One is that it may be hard to evaluate the likelihood function. This might be because the probability model that defines L involves complex processes that are hard to simulate or evaluate. But even when the probability model is simple, L can be expensive to evaluate if the dataset X is large. Optimization algorithms prefer to know derivatives of L with respect to the components of θ. There are specialized optimization algorithms, including those using *adjoint equations*, for evaluating these derivatives. There also are algorithms, including *stochastic gradient descent* and *hierarchical* models that approximate L and $\nabla_\theta L$ when X is large.

Another potential issue is the complexity of $L(x, \theta)$ as a function of θ. In this context, the likelihood function may be called the likelihood *surface* or the *landscape*. This "surface" is the graph of the likelihood function over the parameter space. You can picture it in 3D when the parameter space is two dimensional where $\theta = (\theta_1, \theta_2)$, this surface is the graph of $L(X|\theta)$. Higher dimensional "landscapes" are harder to visualize. From here on, the surface will be turned upside down. Maximizing the likelihood function means finding the top of a hill. Instead, we talk about minimizing the *loss function*, which is the negative of the likelihood function. Minimizing loss may be pictured as looking for the bottom of a valley. You can hope that the landscape is simple and convex like a bowl, but it may turn out to be more like a geological landscape with peaks and valleys (local maxima and minima[1]) winding river valleys (strong non-convexity), etc.

Complex landscapes have many ways to cause trouble. Most optimization algorithms search for the optimizer by going "downhill". Any such algorithm can get stuck in a local minimum that is not the global minimizer. If it is at a local min, all nearby directions are uphill. It is hard to know whether there are better values further away. Landscapes can have so many local minima that the optimization algorithm "gets stuck" in *local wells* (neighborhoods of local minimizers) that are far from being the best fits.

It is often helpful to learn more about the likelihood surface than just the maximum likelihood point $\widehat{\theta}_{\rm ML}$. Nearby parameter combinations fit the data almost as well as $\widehat{\theta}_{\rm ML}$, so they are (almost) as "valid" explanations of the data as the optimal one. You can move a short distance from $\widehat{\theta}_{\rm ML}$ and the fit gets only a little worse. Crucially, how much worse depends on the direction. The shape of the likelihood surface determines something like a *confidence region* (not exactly the correct use of this term) that consists of almost equally valid fits. As I just said, there is good reason to learn the shape of this region, which may be far from spherical. This is one of the main uses of Monte Carlo exploration of the *Bayesian posterior*, which will be discussed later.

The likelihood function also defines the *Fisher information* and the Fisher information *matrix*. A theorem of theoretical frequentist statists relates the Fisher information matrix to the covariance of the random variable $\widehat{\theta}_{\rm ML}(X)$. Recall, this is a random because X is random. The $d \times d$ covariance matrix of the $\widehat{\theta}_{\rm ML}(X)$ contains information about how accurately $\widehat{\theta}_{\rm ML}(X)$ is determined by X. The *Bernstein* theorem of Bayesian statistics relates the Fisher information matrix to the shape of the confidence region.

2 Examples

2.1 Ordinary least squares

Ordinary least squares fitting (usually called simply least squares fitting) chooses parameter[2] $\theta = (\theta_1, \cdots, \theta_d)$ to minimize the sum of the squares of the differences between model predictions and observation data. Suppose a physical model predicts that the "true" but unobserved value of observation k is $f_k(\theta)$. A common probability model is that the observed value X_k is equal to $f_k(\theta)$ with

[1] Terminology: $\overline{\theta}$ is a *local minimum* (more properly, local *minimizer*) if $L(X, \theta) \geq L(X, \overline{\theta})$ when θ is close to $\overline{\theta}$. It is a strict local minimizer if $L(X, \theta) > L(X, \overline{\theta})$ when θ is close to $\overline{\theta}$ but $\theta \neq \overline{\theta}$. It is a *global* minimizer if $L(X, \theta) \geq L(X, \overline{\theta})$ (or for all θ.

[2] From now on, we call the parameter combination "the" parameter. The parameter is a d component object.

some *observation noise*, ϵ_k. The observation error ϵ_k is also called model *residual*. We assume the residuals ϵ_k are gaussian with mean zero and standard deviation σ. We also assume the residuals are independent of each other. "Independent" and "identically distributed" is abbreviated *i.i.d.* The Gaussian distribution, also called *normal*, with mean μ and variance σ^2 is written $\mathcal{N}(\mu, \sigma^2)$. In this notation, the probability model is that the observations are

$$X_k = f_k(\theta) + \epsilon_k \ , \quad \epsilon_k \sim \mathcal{N}(0, \sigma^2) \ \text{ i.i.d.} \tag{2}$$

In this notation, the probability distribution of one data observation X_k is taken to be

$$X_k \sim \mathcal{N}(f_k(\theta), \sigma^2) \ .$$

The corresponding PDF formula for X_k is

$$\begin{aligned} r_k(x_k|\theta) &= N(x_k|f_k(\theta), \sigma^2) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-1}{2\sigma^2}(x_k - f_k(\theta))^2} \ . \end{aligned} \tag{3}$$

The random variables X_k are independent so their joint density is the product of the individual densities. If there are N observations, the PDF of X with parameters θ is (explanation below)

$$L(x_1, \cdots .x_N|\theta) = \prod_{k=1}^{N} r_k(x_k|\theta) \tag{4}$$

$$L(x|\theta) = \left(2\pi\sigma^2\right)^{\frac{-N}{2}} \exp\left(\frac{-1}{2\sigma^2} \sum_{k=1}^{N} (x_k - f_k(\theta))^2 \right) \ . \tag{5}$$

The sum in the exponent, when we put in the actual dataset, is

$$F(X, \theta) = \sum_{k=1}^{N} (X_k - f_k(\theta))^2 \ . \tag{6}$$

This is the sum of the squares of the model fitting residuals for all the data items in the dataset. *Residual* (in computational science) often refers to the amount by which an equation is not satisfied. The model with parameters θ fits data value X_k exactly if $X_k = f_k(\theta)$. The difference between X_k and $f_k(\theta)$ is the residual.

The θ value that maximizes the likelihood (5) is the value that minimizes the sum of squares of F in (6), because of the minus sign in (5). For that reason, least squares minimization is equivalent to maximum likelihood estimation under the hypothesis of i.i.d. gaussian observation errors. The sum of squares (6) is the *loss function* for least squares fitting. Minimizing this loss is equivalent to maximum likelihood.

Here is an explanation of the derivation of (5) from (4) and (3). First, each of the N factors in the product (5) has a factor of $1/\sqrt{2\pi\sigma^2} = \left(2\pi\sigma^2\right)^{-\frac{1}{2}}$. This gives the *prefactor*, $\left(2\pi\sigma^2\right)^{\frac{-N}{2}}$, in (5). Next, you multiply the exponential factors e^{**} in (3) and use the multiplicative property of exponents ($e^a e^b \cdots = e^{a+b+\cdots}$) to get the exponential factor in (5).

2.1.1 Linear least squares

[Warning: There is no good notation or terminology for the material in this material. What's called α below really should be A, for "amplitude". The matrix A is often called the "data matrix" and denoted by X in this context. In machine learning, the rows of A would be called "input variables" and denoted by X_k. The observed valued X_k might be called "labels" and denoted by Y_k. Linear fitting

(and also non-linear fitting) might be called "supervised learning". Giving labels is "supervision" and estimating parameters is "learning". In regression, θ_0 often denotes the coefficient of the constant term. Here, the parameters are $\theta_1, \cdots, \theta_d$. In the example below, θ_3 is the coefficient of the constant term.]

As a more specific example, suppose the observations X_k are positions of an oscillator at time t_k. The observation times t_k are assumed to be known, as well as the oscillator frequency parameter ω. What is unknown are the amplitude of the oscillation, the phase, and the bias. A scalar oscillator with frequency ω, amplitude A, phase angle ϕ and bias $\overline{x}$ produces a signal

$$x(t) = \alpha \sin(\omega t - \phi) + \overline{x} \ .$$

We suppose X_k is a noisy observation of $x(t_k)$:

$$X_k = \alpha \sin(\omega t_k - \phi) + \overline{x} + \epsilon_k \ .$$

The observation errors are modeled as i.i.d. gaussian with variance σ^2 and mean zero. The goal is to estimate α, ϕ, and $\overline{x}$ from the observations X_k and the observation times t_k. The t_k are assumed to be known exactly. In the abstract notation of (2), the parameter set to be estimated is

$$\theta = (\theta_1, \theta_2, \theta_3) = (\alpha, \phi, \overline{x}) \ .$$

The predictive model is

$$f_k(\alpha, \phi, \overline{x}) = \alpha \sin(\omega t_k - \phi) + \overline{x} \ . \tag{7}$$

This formulation gives a nonlinear least squares problem because f_k depends on ϕ in a nonlinear way.

The estimation may be reformulated to be entirely linear. The sine angle addition formula from trigonometry is

$$\sin(a - b) = \sin(a)\cos(b) - \cos(a)\sin(b) \ .$$

Therefore, the model function of (7) may be rewritten as

$$f_k(\alpha, \phi, \overline{x}) = \alpha \cos(\phi) \sin(\omega t_k) - \alpha \sin(\phi) \cos(\omega t_k) + \overline{x} \ .$$

Therefore, we get an equivalent estimation problem by taking the model functions and parameter set to be

$$f_k(\beta, \gamma, \overline{x}) = \beta \sin(\omega t_k) - \gamma \cos(\omega t_k) + \overline{x} \ , \tag{8}$$

$$\theta = (\beta, \gamma, \overline{x}) \ . \tag{9}$$

If we know β and γ, we can recover α and ϕ using formulas from trigonometry

$$\alpha = \left(\beta^2 + \gamma^2 \right)^{\frac{1}{2}} \ , \quad \phi = \operatorname{acos}(\beta/\gamma) \ .$$

The general linear least squares problem involves an $n \times d$ coefficient matrix A and a data vector $X \in \mathbb{R}^n$. Row k of A is $A_k = (a_{k,1}, \cdots, a_{k,d})$. The linear model is

$$X_k = \sum_{j=1}^{d} a_{kj}\theta_j + \epsilon_k \ , \quad \epsilon_k \sim \mathcal{N}(0, \sigma^2) \ , i.i.d \ .$$

As above, for any parameter vector θ, the *residual* of equation k is the amount by which variable X_k is not explained by the model:

$$r_k = \sum_{j=1}^{d} a_{kj}\theta_j - X_k \ .$$

Maximizing the likelihood over θ is the same as minimizing the sum of squares of the residuals (6). Linear estimation problems may be expressed using linear algebra formalism. The residuals r_k form the components of the residual vector

$$r = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix} \quad .$$

The residual vector is given in terms of the coefficient matrix and parameter vector as

$$r = A\theta - X \ .$$

The sum of the squares of the residuals (6) is

$$\sum_{k=1}^{n} r_k^2 = r^T r$$

To summarize, the general linear least squares problem is

$$\min_{\theta} r^T r \ , \quad r = A\theta - X \ . \tag{10}$$

There are specialized methods in numerical linear algebra for solving this.

2.1.2 Nonlinear least squares

Of course, the model functions can be nonlinear. For example, the model in subsection 2.1.1 was nonlinear in the phase angle ϕ before the trigonometry trick. As another example, suppose the frequency ω is unknown. The trick to linearize goes away and (6) becomes the 4 dimensional minimization problem

$$\min_{\omega, A, \phi, \overline{x}} \sum_{k=1}^{n} \left(X_k - A\sin(\omega t_k - \phi) - \overline{x} \right)^2 \ .$$

We will see later that the best solution algorithms for problems like this may take advantage of the fact that many of the parameters appear linearly. Minimizing over the linear problems is more explicit and leaves a minimization problem only over the few "nonlinear" parameters (parameters that appear nonlinearly). In this example, we would reduce the "nonlinear dimension" from 2 to 1 by using the trigonometry trick.

The model functions $f_k(\theta)$ in (6) might be too complicated to be given by explicit formulas. As an example of that, suppose the problem is to *callibrate* ("learn") a nonlinear dynamical model from data. A simple model might be a nonlinear oscillator

$$\ddot{x} = -\omega_0^2 x - \gamma\dot{x} - \beta x^3 + r\sin(\omega_1 t) \ . \tag{11}$$

The parameters are:

ω_0:	small amplitude oscillation frequency	to be learned
γ:	coefficient of friction	to be learned
β:	coefficient of nonlinearity	to be learned
r:	amplitude of driving force	assumed known
ω_1:	frequency of driving force	assumed known

There are measurements $X_k = x(t_k)$. There is no formula for the solution of the dynamical model (11). Instead, the model predictions are found by simulating the oscillator equations numerically. As before, $f_k(\theta) = f_k(\omega_0, \gamma, \beta)$, but now the value of f_k is computed by numerical simulation rather than

an explicit formula. Optimization algorithms for the minimization often use derivatives $\nabla_\theta f_k(\theta)$. This *sensitivity analysis* may be done either by differentiating the dynamical equations (see below) or by using automatic differentiation software. As a last resort (rare, these days), the derivatives may be estimated by numerical finite differences.

Multiple local minima are a common feature of nonlinear models. A linear least squares problem has a unique solution as long as the coefficient matrix A in (10) has full rank. Nonlinear models lead to the problem of minimizing a function F in (6) that is not quadratic and is likely to have more than one local minimizer. For example, the fitting problem in subsection 2.1.1 has multiple local minima if you use the phase angle ϕ. The fitting error is the same for ϕ and $\phi + 2\pi$, etc. Thus, for any local minimizer there is an infinite sequence of equivalent local minimizers with phase $\phi' = \phi + 2\pi j$. These minimizers are equivalent physically, but they are not equivalent mathematically and are not equal algorithmically. Other nonlinear models have local minimizers that are not equivalent physically. The exercises have an example of this.

2.2 Non-gaussian probability models

There are many probability distributions that are not Gaussian. Sometimes we choose a distribution because there is a physical reason for it. Sometimes we choose a distribution because it has properties we want to include in a model, even though there is not a compelling physical argument for that specific distribution.

2.2.1 Exponential and Poisson

The exponential distribution models, among other things, the time it takes something (a lightbulb, perhaps) to break. The model is that breaking is a freak occurrence not related to the age of the thing. Let $T > 0$ be the time it breaks. There is a *rate parameter*, λ, that determines the probability of the freak event in a small interval of time. Specifically, if $T > t$ and dt is small, the

$$\Pr(\, t < T < t + dt \mid T > t \,) = \lambda \, dt \; .$$

Bayes' rule of conditional probability allows this to be rewritten as

$$\frac{\Pr(\, t < T < t + dt \text{ and } t < T \,)}{\Pr(\, t < T \,)} = \lambda \, dt \; .$$

Obviously (because $t < T < t + dt$ implies $t < T$),

$$\Pr(\, t < T < t + dt \text{ and } t < T \,) = \Pr(\, t < T < t + dt \,) \; .$$

Therefore, the model may be written as

$$\Pr(\, t < T < t + dt \,) = \lambda \, dt \Pr(\, t < T \,) \; .$$

Now let $p(t)$ be the PDF of T and $P(t)$ be the CDF. For $t > 0$, the relationship between P and p may be written in either of the forms

$$P(t) = \Pr(\, T < t \,) = \int_0^t p(s) \, ds$$

$$P'(t) = p(t) \quad , \quad P(0) = 0 \; .$$

The definition of probability density is (for small enough dt)

$$\Pr(\, t < T < t + dt \,) = p(t) \, dt \; .$$

Thus, we get

$$p(t)\,dt = \lambda\,dt\,P(t) \quad , \quad \text{or} \quad P'(t) = \lambda\,P(t) \ .$$

With $P(0) = 0$, this gives the exponential distribution as (for $t > 0$)

$$P(t) = 1 - e^{-\lambda t}$$
$$p(t) = \lambda e^{-\lambda t} \ .$$

The distribution is called *exponential* because of formulas for p and P. We may write $T \sim \exp(\lambda)$ to express the idea that T is an exponential random variable with rate parameter λ.

As an example, suppose a model predicts that event k happens at time $t_k = f_k(\theta)$ but is observed only after an exponential time $T_k \sim \exp(\lambda)$. Then the PDF for the observation time of observation k is

$$r_k(t_k|\lambda,\theta) = \begin{cases} 0 & \text{if } t < f_k(\theta) \\ \lambda e^{-\lambda(t - f_k(\theta))} & \text{if } t > f_k(\theta) \end{cases}$$

The likelihood function for n dataset $T = T_1, \cdots, t_n)$ is

$$L(T|\lambda,\theta) = \begin{cases} \exp\left(-\lambda \sum_{k=1}^{n} (\,T_k - f_k(\theta)\,) \right) & \text{if } T_k > f_k(\theta) \text{ for all } k \\ 0 & \text{if } T_k < f_k(\theta) \text{ for any } k \end{cases} \tag{12}$$

If λ is known, then maximum likelihood estimation of θ involves optimizing (12) over θ. Otherwise, we would optimize over the $d+1$ variables (λ, θ). Either way, the optimization problem is challenging because the parameters θ are restricted to the *feasible set* where all the inequalities $T_k > f_k(\theta)$ are satisfied.

The Poisson random variable is a non-negative integer N that represents the number of "freak events" (sometimes called *arrivals*) of an exponential with rate λ over a time t. A probability book will explain that the distribution depends on a parameter $\mu = \lambda t$

$$\Pr(\,N = n\,) = p_n(\mu) = \frac{\mu^n}{n!} e^{-\mu} = \frac{(\lambda t)^n}{n!} e^{-\lambda t} \ .$$

Suppose we have a dataset consisting of m observations N_k over time intervals of length t_k and we want to estimate λ. The relevant likelihood function is

$$L(N|\lambda) = \frac{t_1^{N_1} \cdots t_m^{N_m}}{N_1! \cdots N_m!}\, \lambda^{N_1 + \cdots + N_m}\, e^{-\lambda(t_1 + \cdots + t_m)} \ .$$

You can find the maximum of L over λ by calculus. The result is

$$\lambda_{\text{MLE}} = \frac{N_1 + \cdots + N_m}{t_1 + \cdots + t_m} \ .$$

This formula has the simple interpretation of the total number of "hits" divided by the total observation time. This might seem obvious in retrospect.

2.2.2 Fat-tailed t

The Gaussian distribution is convenient for closed form solutions. That's why people are temped to assume observation errors are Gaussian even when they don't really know. Aside from being wrong (which may not be so terrible by itself), assuming Gaussian observation errors makes statistical estimates vulnerable to *outliers*. An outlier is a piece of data that is far from a pattern, either because of a data recording error or because the process itself produces occasional outliers. An estimation method is called *robust* if it can work even when there are outliers, for whatever reason.

One way to make a statistical estimator robust is to model observation errors with a probability distribution that allows outliers. These are called *fat tailed* distributions. The *tails* of a probability distribution are the parts many standard deviations from the mean. You can gauge the size of the tails using the probability of being k standard deviations from the mean. For a Gaussian, these probabilities decrease very quickly as k gets large. This can be seen from the approximate formula[3]

$$\Pr_{\mathcal{N}}(X > \mu + k\sigma) \approx \frac{1}{\sqrt{2\pi}} \frac{1}{k} e^{-\frac{1}{2}k^2} \ . \tag{13}$$

The subscript $\mathcal{N}$ means normal (gaussian). The mean and variance are not given because they are irrelevant. The approximation estimates

$$\begin{aligned} \Pr_{\mathcal{N}}(X > \mu + 2\sigma) &\approx .027 \\ \Pr_{\mathcal{N}}(X > \mu + 3\sigma) &\approx .0015 \\ \Pr_{\mathcal{N}}(X > \mu + 5\sigma) &\approx 3{\cdot}10^{-7} \ . \end{aligned}$$

These estimates are reasonably accurate, as the exact numbers are .0275, .00135, and $\approx 2.9 \cdot 10^{-7}$ respectively. To appreciate the smallness of these tails, the probability of having a "5σ" event one day in a lifetime (365 days $\times 100$ years $= 3.6 \cdot 10^4$ days) is about $3 \cdot 10^{-7} \times 3.6 \cdot 10^4 \approx .011 = 1.1\%$. Yet, there have been several 5σ days on the stock market (stock prices drop that much in a day) in my lifetime.

The *Student t* distribution is a distribution that is approximately gaussian in the central region but has fatter tails (larger tail probabilities). The PDF for a "standard" t random variable with ν "degrees of freedom" is

$$p(x) = \frac{1}{Z} \left(1 + \frac{x^2}{\nu} \right)^{-\frac{\nu+1}{2}} \ . \tag{14}$$

The normalization constant Z may be found in books, but it is often irrelevant. You can check that t is approximately gaussian for large ν (many *degrees of freedom*, because

$$\left(1 + \frac{t}{n} \right)^{-n} \approx e^{-t} \ , \quad \text{take } n = \frac{\nu+1}{2} \ , \ \text{ so } \nu \approx 2n \ .$$

The probability of a 3σ event with $\nu = 3$ degrees of freedom is about .0069 and the probability of a 5σ event is about .0016. These probabilities are far larger than their gaussian equivalents. You would expect a 5σ event in less than two years rather than more than a hundred years.

You can make a t random variable with mean μ and length scale r as you do for Gaussians. Note that $r = 1$ (the "standard" t) has variance σ^2 a little larger than 1.

$$p_{\mu,r} = \frac{1}{Z(r)} \left(1 + \frac{(x-\mu)^2}{\nu r^2} \right)^{-\frac{\nu+1}{2}} \ .$$

If you need to know how the normalization constant depends on r, you can see that $Z(r) = Z_1 r$ and

$$p_{\mu,r} = \frac{1}{Z_1 r} \left(1 + \frac{(x-\mu)^2}{\nu r^2} \right)^{-\frac{\nu+1}{2}} \ . \tag{15}$$

The formulas for the partial derivatives $\partial_\nu p$, $\partial_r p$ and $\partial_\mu p$ may now be found explicitly, though the results may not seem beautiful.

[3]The approximation (13) has a name but I don't know it. To derive it, assume (as you may) $\mu = 0$ and $\sigma = 1$. Then $\Pr_{\mathcal{N}}(X > \mu + k\sigma) = \frac{1}{\sqrt{2\pi}} \int_k^\infty e^{-\frac{1}{2}x^2} dx$, make the substitution $x = k + y$ and get $\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}k^2} \int_0^\infty e^{-ky - \frac{1}{2}y^2} dy$. Finally, leave out $\frac{1}{2}y^2$ in the integral. The integral that remains is $\int_0^\infty e^{-ky} dy = \frac{1}{k}$. The approximation is justified because y values where $\frac{1}{2}y^2$ matters have $ky + \frac{1}{2}y^2$ so large that e^{-**} contributes little to the integral.

2.3 Mixture models

Mixture models are appropriate when different elements of the dataset are produced in different ways but the mechanism producing the observation is not itself observed. A textbook example[4] involves the weights of N fish that may be either type 1 or type 2 (possibly female and male, possibly related species). The weight of a type 1 fish is gaussian $\mathcal{N}(\mu_1, \sigma_1^2)$, while a type 2 fish has weight $\mathcal{N}(\mu_2, \sigma_2^2)$. In the mixture model, a fish is type 1 with probability p_1 and type 2 with probability p_2. Let $r_j(x)$ be the PDF of weights of fish of type j (both Gaussian in this example). This is easily extended to more than two types. We simplify notation for the case of two types by writing $p_1 = p$ and $p_2 = 1-p$. The overall PDF of a fish weight is

$$\begin{aligned} q(x) &= r_1(x)\,\Pr(\text{ fish is type 1 }) + r_2(x)\,\Pr(\text{ fish is type 2 }) \\ &= p_1\, r_1(x) + p_2\, r_2(x)\ . \\ &= pN(x|\mu_1, \sigma_1^2) + (1-p)N(x|\mu_2, \sigma_2^2) \\ q(x|\mu_1, \mu_2, \sigma_1, \sigma_2, p) &= p\frac{1}{\sqrt{2\pi\sigma_1^2}}\, e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} + (1-p)\frac{1}{\sqrt{2\pi\sigma_1^2}}\, e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}\ . \end{aligned} \tag{16}$$

The parameter θ has five components $\theta = (\mu_1, \mu_2, \sigma_1, \sigma_2, p)$.

2.4 Log likelihood

The *log likelihood* function is the log of the likelihood function. It is often denoted as $\ell(\theta)$, with the dependence on the data omitted:

$$\ell(\theta) = \log(\, L(X|\theta)\,)\ .$$

For likelihood optimization (maximum likelihood estimation), it does not matter which one you use, since

$$\widehat{\theta}_{\text{MLE}} = \arg\ \max_\theta \ell(\theta) = \arg\ \max_\theta L(X|\theta)\ .$$

Still, there are practical and theoretical advantages to log likelihood over likelihood.

One practical advantage concerns computer arithmetic. This will be discussed in more detail later. For now, the largest single precision (32 bit) floating point number is about 2^{127}. If each factor in the likelihood product (4) is at least 2, and if there are at least 127 factors (corresponding to at least 127 datapoints), then the product will be larger than 2^{127} and therefore cannot be represented as a simple floating point number in the computer. Taking the log helps because it transforms the product into the sum

$$\ell(\theta) = \sum_{k=1}^{N} \log(\, r_k(X_k, \theta)\,)\ .$$

Now if there are $N = 1000$ terms in the sum and each one is less than $\log(10) < 3$, then ℓ is less than 3000, which is well within the range of single precision floating point arithmetic.

A second practical advantage concerns numerical optimization. The likelihood function tends to be "flat" (have a small gradient) for very poor fits. Optimization algorithms often measure convergence by the size of the gradient. This means, for example, that a gradient descent iteration (described later) might stop if it found θ_* with $\|\nabla_\theta L(X|\theta)\| < 10^{-10}$. However, this might happen if θ_* is far from $\widehat{\theta}_{\text{MLE}}$ because of the exponentials in L. As a simple example, suppose there is one datapoint X a gaussian model with $\sigma = 1$. Then $L(X|\mu) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(X-\mu)^2}$ and the gradient is

$$\partial_\mu L = -\frac{1}{\sqrt{2\pi}}(X-\mu)e^{-\frac{1}{2}(X-\mu)^2}\ .$$

[4] I don't know whether it's real.

This has $|\partial_\mu L(X|\mu)| < 10^{-10}$ if $X - \mu = 7$. This is far from the MLE value $\widehat{\mu}_{\rm MLE} = X$. On the other hand

$$\ell(\mu) = -\frac{1}{2}\log(2\pi) - \frac{1}{2}\,(\,X-\mu\,)^2 \ .$$

The constant term $\frac{1}{2}\log(2\pi)$ is irrelevant for optimization (as was the multiplicative prefactor it is the log of). You can see that the gradient becomes larger, not smaller, as μ gets farther away from X. As a consequence, in this example at least, and even if computer arithmetic were done exactly for arbitrarily large or small numbers, optimizing ℓ is easier than optimizing L,

A third practical advantage is that differentiating a big sum like ℓ can be easier than differentiating a big product like L. In the Gaussian noise case, for example, differentiating (6) is simpler than differentiating (5).

As a final application, we use logarithmic differentiation to see how the fat-tailed t model makes estimation less sensitive to outliers. Suppose $X = (X_1, \cdots, X_N)$ is a dataset of N numbers that are modeled as independent measurements of an underlying number μ. The noise model is

$$X_k = x_* + \epsilon_k \quad , \quad \epsilon_k \sim {\rm T}(\nu, 1, 0) \ .$$

The notation ${\rm T}(\nu, r, \mu)$ is for the t random variable ν degrees of freedom, scale variable r and centered at μ. The PDF is given by (15). Suppose ν and r are fixed and we only seek the maximum likelihood estimate of x_*. The likelihood function in this model is

$$L = \frac{1}{Z_1^N r^N}\prod_{k=1}^N \left(\,1 + \frac{(X_k-\mu)^2}{\nu r^2}\,\right)^{-\frac{\nu+1}{2}} \ .$$

The log likelihood is

$$\ell(\mu) = -N(\,\log Z_1 + \log(r)\,) - \frac{\nu+1}{2}\sum_{k=1}^N \log\left(\,1 + \frac{(X_k-\mu)^2}{\nu r^2}\,\right) \ .$$

We find the maximum likelihood $\widehat{\mu}_{\rm MLE}$ by differentiating and setting the derivative equal to zero. The derivative is (check this)

$$\partial_\mu\,\ell(\mu) = \frac{\nu+1}{\nu r^2}\sum_{k=1}^N \frac{1}{1+\frac{(X_k-\mu)^2}{\nu r^2}}\ (\,X_k - \mu\,) \ .$$

The prefactor is irrelevant when setting $\partial_\mu \ell(\mu) = 0$. We define positive *weights*,

$$w_k = \frac{1}{1+\frac{(X_k-\mu)^2}{\nu r^2}} \ . \tag{17}$$

The weight w_k is largest when $X_k = \mu$ and decreases as X_k moves away from μ. The sum of the weights will be a normalization factor

$$Z = \sum_{k=1}^n w_k \ .$$

In this notation:

$$\partial_\mu\,\ell(\mu) = 0 \implies \mu = \frac{1}{Z}\sum_{k=1}^N w_k X_k \ . \tag{18}$$

This shows that the optimal μ is a weighted average of the measurements X_k. The measurements X_k close to μ have large w_k while those far away have small weight. You can see the effect of this by imagining what happens if $X_1 \to \infty$ (because of some kind of data error). In this case $w_1 \to 0$ and

the influence of X_1 decreases to zero. In the limit, μ will be determined only by the remaining X_k with $k \neq 1$. This makes the maximum likelihood estimate of μ *robust* against a few outliers.

Maximum likelihood for this case is not as simple as this might make it seem. The formula (18) is not an explicit formula for $\widehat{\mu}_{\mathrm{MLE}}$ because the weights also depend on μ. The equation (18) must be solved using some iteration procedure of the kind we will talk about later in the class. An exercise in Assignment 2 will show that there may be more than one solution. "The" maximum likelihood estimate may not be unique (there may be more than one local minimizer).