Cosmσlogy

Contents

Part I
The Fourier Transform

1 Square Integrable Functions

Square integrable functions are an extremely important part of quantum mechanics and many branches of Physics. They allow the easy construction of function "inner products" – or a Hilbert space – and the Born probability amplitude $|\psi|^2$.

1.1 P-Norms

For the purposes of this section, we will restrict our discussion only to square integrable functions in one dimensional euclidean space, or $L^2(E_1)$. These functions have the propertry

$$(\int_{-\infty}^{\infty} |f(x)|^2 dx)^{1/2} < \infty, \tag{1}$$

Which means the "two norm" of f converges. A general p norm is defined as

$$\|f\|_p \equiv (\int_{-\infty}^{\infty} |f(x)|^p dx)^{1/p} \tag{2}$$

We define $|f|_\infty$ to be the supremum of the function f, or the highest value it attains over its domain. You will see shortly that this notion of a p norm is tightly connected with the abstract idea of an "inner product" between two functions.

1.2 F(s) and f(x)

The Fourier Transform is a bounded linear transform which maps $f \to F$. We define the transform as

$$F(s) = \int f(x) e^{2\pi i x s} dx \tag{3}$$

The resulting function F, can be synonymously referred to as $\hat{f}$ or $\bar{f}$. We will make the definition clear by context. Here are a few compact ways to write the transformation above:

$$\begin{aligned} f &\supset F && (4)\\ \bar{f} &= F && (5)\\ \hat{f} &= F && (6) \end{aligned}$$

Note that, there is a concept of a "forward" and a "backward" transform, in the sense that one takes you "into" frequency space, and one takes you "out" of frequency space:

$$\begin{aligned} \overline{f(x)} = F(s) &= \int f(x) e^{-2\pi i x s} dx && (7)\\ \overline{F(s)} = f(x) &= \int F(s) e^{2\pi i s x} ds && (8) \end{aligned}$$

This first is a "forward" transform – which we associate with a negative argument in the exponential; the second is a "backward" transform – which we associate with a positive argument in the exponential. Notice that, if we forward transform twice:

$$f(x) \supset F(k) \supset f(-x), \tag{9}$$

but

$$f(x) \supset F(k) \text{ ``forwards''} \tag{10}$$
$$f(x) \subset F(k) \text{ ``backwards''.} \tag{11}$$

The Fourier transform therefore, has a well defined inverse and we say that f and F form a transform pair. In $L^2(E_1)$ theory, we can say without doubt that the backwards Fourier transform is finite, or that the integral:

$$f(x) = \int F(k)e^{ikx}dx \tag{12}$$

converges. This is not necessarily true in $L^1(E_1)$ theory.

Many physically-oriented people like to think of $f(x)$ as a superposition of complex waves e^{ikx}, with the component function $F(k)$. Some people even like to think of $f(x)$ as a vector – even though its not – built out of orthogonal wave components e^{ikx}. Such notions of Fourier analysis can be helpful in quantum mechanics.

1.3 Inner Products and the Hilbert Space

We can treat square integrable functions, say $f(x)$ and $g(x)$ as vectors in an inner product space. This is because the inner product

$$\langle f, g\rangle = \int f(x)g(x)dx \tag{13}$$

is well defined – i.e. never reaches infinity or meaningless values. This is because both f and g were square integrable, $\|f\|_2 < \infty$, $\|g\|_2 < \infty$. Let us prove this with the a triangle identity of sorts. First we examine the inner product of f with itself, or, its 2 norm.

$$\langle f, f\rangle = \int f(x)^2dx \tag{14}$$
$$\langle f, f\rangle = \int |f(x)|^2dx \tag{15}$$
$$\sqrt{\langle f, f\rangle} = \sqrt{\int |f(x)|^2dx} \tag{16}$$
$$\sqrt{\langle f, f\rangle} = \|f\|_2 \tag{17}$$

Next, we look at the square of the inner product between f and g

$$\begin{aligned}\langle f,g\rangle &= \int f(x)g(x)dx && (18)\\ |\langle f,g\rangle|^2 &= (\int f(x)g(x)dx)^2 && (19)\\ |\langle f,g\rangle|^2 &\leq \int |f(x)g(x)|^2dx && (20)\\ |\langle f,g\rangle|^2 &\leq \int |f(x)|^2|g(x)|^2dx && (21)\\ |\langle f,g\rangle|^2 &\leq \int |f(x)|^2dx\int |g(x)|^2dx && (22)\\ |\langle f,g\rangle|^2 &\leq \|f\|^2\|g\|^2 && (23)\end{aligned}$$

Where, we have excised the subscript 2 for the "two" norm to avoid confusion with the squared power. Notice that this inner product is bounded between f and g. We have a well defined scalar, created from this inner product of two functions! This relation is called the Cauchy-Schwartz inequality, and is intimately related with the dot product in cartesian space between two vectors:

$$\begin{aligned}\vec{a}\cdot\vec{b} &= |a||b|\cos\theta && (24)\\ |\vec{a}\cdot\vec{b}|^2 &= |a|^2|b|^2(\cos\theta)^2 && (25)\\ |\vec{a}\cdot\vec{b}|^2 &= (\vec{a}\cdot\vec{a})(\vec{b}\cdot\vec{b})(\cos\theta)^2 && (26)\\ \vec{a}\cdot\vec{b} &\leq |a||b| && (27)\end{aligned}$$

This space of square-integrable $L^2(E_1)$ functions is a Hilbert Space: a real or complex inner product space that is also a complete metric space. The latter part of that sentence we won't get into at the moment; the main takeaway is to understand that two functions can now be deemed orthogonal if their inner product is zero. Take $\sin x$ and $\cos x$ for example, or any even function f and odd function g: if these two functions are both square integrable, we find that their inner product is zero! (Note however, that for $\sin$ and $\cos$, we require our integration bounds to be finite; say $-\pi/2 \to \pi/2$.)

1.4 Inner product of two Sinusoids

It is easily shown that the inner product of two sinusoidal functions is zero unless they carry the same frequency argument. The fundamental assumption in Fourier analysis is that we can build *any* function out of sinusoids – which are a complete, orthogonal set of basis "vectors". We often write the expansion of a function f in terms of sinusoids of period L:

$$f(x) = a_0 + \sum a_n \cos(\frac{2\pi nx}{L}) + b_n \sin(\frac{2\pi nx}{L}) \qquad (28)$$

Using trigonometric identities, it is easily shown that

$$\int_0^L \cos(\frac{2\pi mx}{L})\sin(\frac{2\pi nx}{L})dx = 0 \;\; \forall m,n$$
$$\int_0^L \cos(\frac{2\pi mx}{L})\cos(\frac{2\pi nx}{L})dx = 0 \;\; m\neq n$$
$$\int_0^L \sin(\frac{2\pi mx}{L})\sin(\frac{2\pi nx}{L})dx = 0 \;\; m\neq n$$

For the case $m = n$, we find

$$\begin{aligned}\int_0^L \cos(\frac{2\pi nx}{L})\cos(\frac{2\pi nx}{L})dx = \frac{L}{2}\\ \int_0^L \sin(\frac{2\pi nx}{L})\sin(\frac{2\pi nx}{L})dx = \frac{L}{2}\end{aligned}$$

If we would like to correctly write any function f in terms of sinusoids, we need to calculate the a_n and b_n terms. This is easily accomplished by something called "Fourier's" trick:

$$\begin{aligned}f(x) &= a_0 + \sum a_n \cos(\frac{2\pi nx}{L}) + b_n \sin(\frac{2\pi nx}{L})\\ \int f(x)\sin(\frac{2\pi mx}{L}) &= \int a_0 \sin(\frac{2\pi mx}{L})dx +\\ &\int \sum a_n \cos(\frac{2\pi nx}{L})\sin(\frac{2\pi mx}{L})dx +\\ &\int \sum b_n \sin(\frac{2\pi mx}{L})\sin(\frac{2\pi nx}{L})dx\end{aligned}$$

The integral of a sum is equal to the sum of the integrals, and we know that all the mixed $\cos$ and $\sin$ terms will go to zero. Leaving us with:

$$\int f(x)\sin(\frac{2\pi nx}{L}) = \int \sum b_n \sin(\frac{2\pi nx}{L})\sin(\frac{2\pi mx}{L})dx.$$

Which will be zero unless we set $m = n$. Meaning

$$\begin{aligned}\int_0^L f(x)\sin(\frac{2\pi nx}{L}) &= b_n\frac{L}{2}\\ \frac{2}{L}\int_0^L f(x)\sin(\frac{2\pi nx}{L}) &= b_n\end{aligned}$$

Conversely, we find the same for $\cos$ and the a_n terms:

$$\begin{aligned}\int_0^L f(x)\cos(\frac{2\pi nx}{L}) &= a_n\frac{L}{2}\\ \frac{2}{L}\int_0^L f(x)\cos(\frac{2\pi nx}{L}) &= a_n\end{aligned}$$

1.5 Even and Odd Functions

Normally in Physics, we are interested in the Fourier transform of a real function. $F(k)$ is obviously complex-valued, and we note the above definition has some implications on the evenness, oddness, and realness of our transform. Let us first examine the complex conjugate of our transform F

$$\begin{aligned}
F(s) &= \int f(x)e^{2\pi isx}dx && (29)\\
F^{\star}(s) &= \int f(x)e^{-2\pi isx}dx && (30)\\
F^{\star}(-s) &= \int f(x)e^{2\pi isx}dx && (31)\\
F^{\star}(-s) &= F(s) && (32)
\end{aligned}$$

Functions with this property are called Hermitian. Or notice that, whenever we took the complex conjugate of F, we did not change the sign of $f(x)$ – in fact we didn't change anything about the function, because we assumed it was purely real (i.e. $f^{\star}(x) = f(x)$. If f is real, then F is complex Hermitian. Now let us split up the complex exponential into its real and imaginary parts:

$$\begin{aligned}
F(s) &= \int f(x)e^{2\pi isx}dx && (33)\\
F(s) &= \int f(x)\cos(2\pi sx)dx - i\int f(x)\sin(2\pi sx)dx && (34)
\end{aligned}$$

First, note that any function f can be *uniquely* described by the superposition of an even and and odd function – $f(x) = E(x) + O(x)$. To prove this, let us split up our $f(x)$ into two separate even and odd pairs:

$$\begin{aligned}
f(x) &= E_1(x) + O_1(x)\\
f(x) &= E_2(x) + O_2(x)\\
E_1(x) + O_1(x) &= E_2(x) + O_2(x)\\
E_1(x) - E_2(x) &= O_1(x) - O_2(x)
\end{aligned}$$

Our final result compares an completely even function (left hand side of the equation) to a completely odd function (right hand side). This is impossible. And so we find the only solution to this equation is the not-so-trivial case $E_1 = E_2$, $O_1 = O_2$. We can now claim that

$$\begin{aligned}
F(s) &= \int f(x)e^{2\pi isx}dx && (35)\\
F(s) &= \int E(x)\cos(2\pi sx)dx - i\int O(x)\sin(2\pi sx)dx && (36)
\end{aligned}$$

Notoce that if $f(x)$ is a completely even function $f(x) = E(x)$, the second integral is zero and the transform is *completely real*. Obversely, if $f(x)$ is a completely odd function, then we find that our transform is completely imaginary. Furthermore, $F(s)$ inherits the even and odd properties of $f(x)$. Let's show this by flipping the sign of x for the even and odd case. In the even case, our second integral is zero and we find:

$$\begin{aligned}
\overline{f(-x)} &= \int E(-x)\cos(2\pi s(-x))dx - i\int O(-x)\sin(2\pi s(-x))dx && (37)\\
&= \int E(x)\cos(2\pi(-s)x)dx - 0 && (38)\\
&= \int f(x)\cos(2\pi s'x)dx && (39)\\
&= F(s') && (40)\\
\Rightarrow F(s) &= F(-s) && (41)
\end{aligned}$$

In the odd case, our first integral is zero and we find:

$$\begin{aligned}
\overline{f(-x)} &= \int E(-x)\cos(2\pi s(-x))dx - i\int O(-x)\sin(2\pi s(-x))dx && (42)\\
&= -\int O(x)\sin(2\pi(-s)x)dx && (43)\\
&= -\int O(-x)\sin(2\pi s' x)dx && (44)\\
&= \int O(x)\sin(2\pi s' x)dx && (45)\\
\Rightarrow F(s) &= -F(-s) && (46)
\end{aligned}$$

If f is even, its transform F is even; if f is odd, its transform F is odd. Summarizing:

1. If $f(x)$ is even and real, then $F(k)$ is even and real.
2. If $f(x)$ is odd and real, then $F(k)$ is odd and imaginary.
3. If $f(x)$ is even and imaginary, then $F(k)$ is even and imaginary.
4. If $f(x)$ is odd and imaginary, then $F(k)$ is odd and real.

Note that points (1) and (2) for real valued function f, imply $F^*(-k) = F(k)$, or that the complex conjugate of the transform is equal to the reflected transform. Such function $F(k)$ – or matrices for that matter – are called Hermitian.

1.6 Dilation Differentiation Translation

There are quite a few nice properties to $F(k)$, let us tabulate them briefly:

1. Dilation

$$\begin{aligned}
\overline{f(ax)} &= \int f(ax)e^{-2\pi isx}dx && (47)\\
&= \frac{1}{|a|}\int f(x)e^{-2\pi i\frac{s}{a}x}dx && (48)\\
&= \frac{1}{|a|}F(\frac{s}{a}) && (49)
\end{aligned}$$

2. Translation

$$\begin{aligned}
\overline{f(x-a)} &= \int f(x-a)e^{-2\pi isx}dx && (50)\\
&= \int f(x')e^{-2\pi is(x'+a)}dx' && (51)\\
&= F(s)e^{-2\pi isa} && (52)
\end{aligned}$$

3. Differentiation

$$\begin{aligned}\frac{\partial F(s)}{\partial s} &= \frac{\partial}{\partial s}\int f(x)e^{-2\pi isx}dx && (53)\\ &= \int -2\pi ixf(x)e^{-2\pi isx}dx && (54)\\ \frac{\partial F(s)}{\partial s} &= \overline{-2\pi ixf(x)} && (55)\\ \frac{i^n}{(2\pi)^n}\frac{\partial F(k)}{\partial k} &= \overline{x^n f(x)} && (56)\end{aligned}$$

1.7 Convolution

We can say that the L^2 space, apart from being an inner product space, is also equipped with a multiplication operation. We can "combine" two square-functions through convolution, an operation labeled by $f * g$ for arbitrary functions of f and g and defined as

$$\int f(x)g(y-x)dx = (f*g)(y) \qquad (57)$$

Notice that the final result is only a function of y. x has been "integrated away". This operation can be understood in the following way: take $g(x)$ and flip it about the y axis – i.e. turn it around; now displace that flipped function g from the origin by varying lengths y. Run it along the x-axis and measure the area overlap between g and the non-displaced function f by integrating over the multiplicative heights at each $x + \delta x$.

Some people even think of convolution as "melting" one function's height into the form of another and adding up this "melting" operation for every possible displacement y. For example, a very noisy function – let's call f white noise – convolved with a Gaussian g will yield a much smoother result $f * g$. One could claim that the sharp peaks of the white noise function f were melted and $smoothed$ into the form of a Gaussian g at each and every x value. For example, smoothing or "blurring" of images normally takes the form of convolution with a Gaussian:

Convolution of two functions is an incredibly important concept – and easily digestible in Fourier space because of the convolution theorem (1). Convolution describes: the probability density function of the observable $Z = X + Y$ the sum two mutually independent random variables X and Y; the smearing effects of data associated with lab equipment; atomic scattering factors; diffraction from a lattice (or a crystal); and how to fill in missing data for known probability distributions in k-space.

Theorem 1. *Let $f(x)$ and $g(x)$ be square integrable functions; the Fourier transform of the convolution is product of the transforms. Namely, $\overline{(f*g)} = FG$ and vice versa, $\overline{F*G} = fg$.*

Proof. Let us examine the convolution of f and g:

$$\int f(x)g(y-x)dx = (f*g)(y), \qquad (58)$$

and then transform that convolution into Fourier space,

$$\begin{aligned}\int\int f(x)g(y-x)e^{-2\pi isy}dxdy &= \overline{(f*g)(y)} && (59)\\ \int\int f(x)g(y-x)e^{-2\pi isy}dydx &= \overline{(f*g)(y)} && (60)\\ \int f(x)G(s)e^{-2\pi isx}dx &= \overline{(f*g)(y)} && (61)\\ F(s)G(s) &= \overline{(f*g)(y)}. && (62)\end{aligned}$$

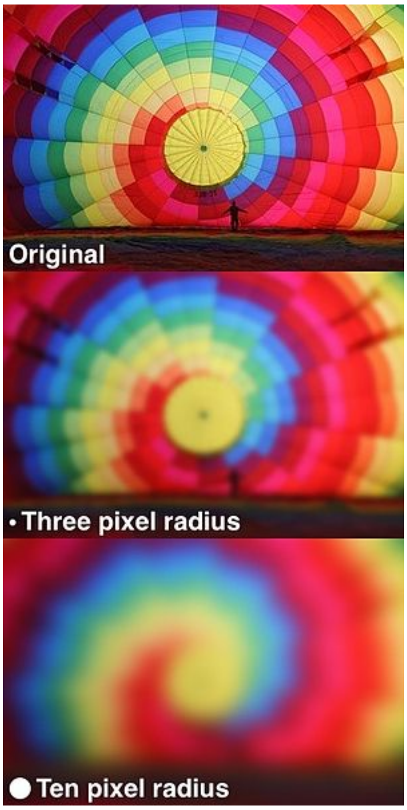

Figure 1: A very pretty picture convolved with a two dimensional Gaussian function, whose smoothing length or variance is equal to various pixel values. In its function form, this smoothing function probably looks like $G(x) \sim e^{(x^2+y^2)/(2N)}$, where N is the number of pixels, or the variance of the Gaussian, and $x^2 + y^2 = r^2$.

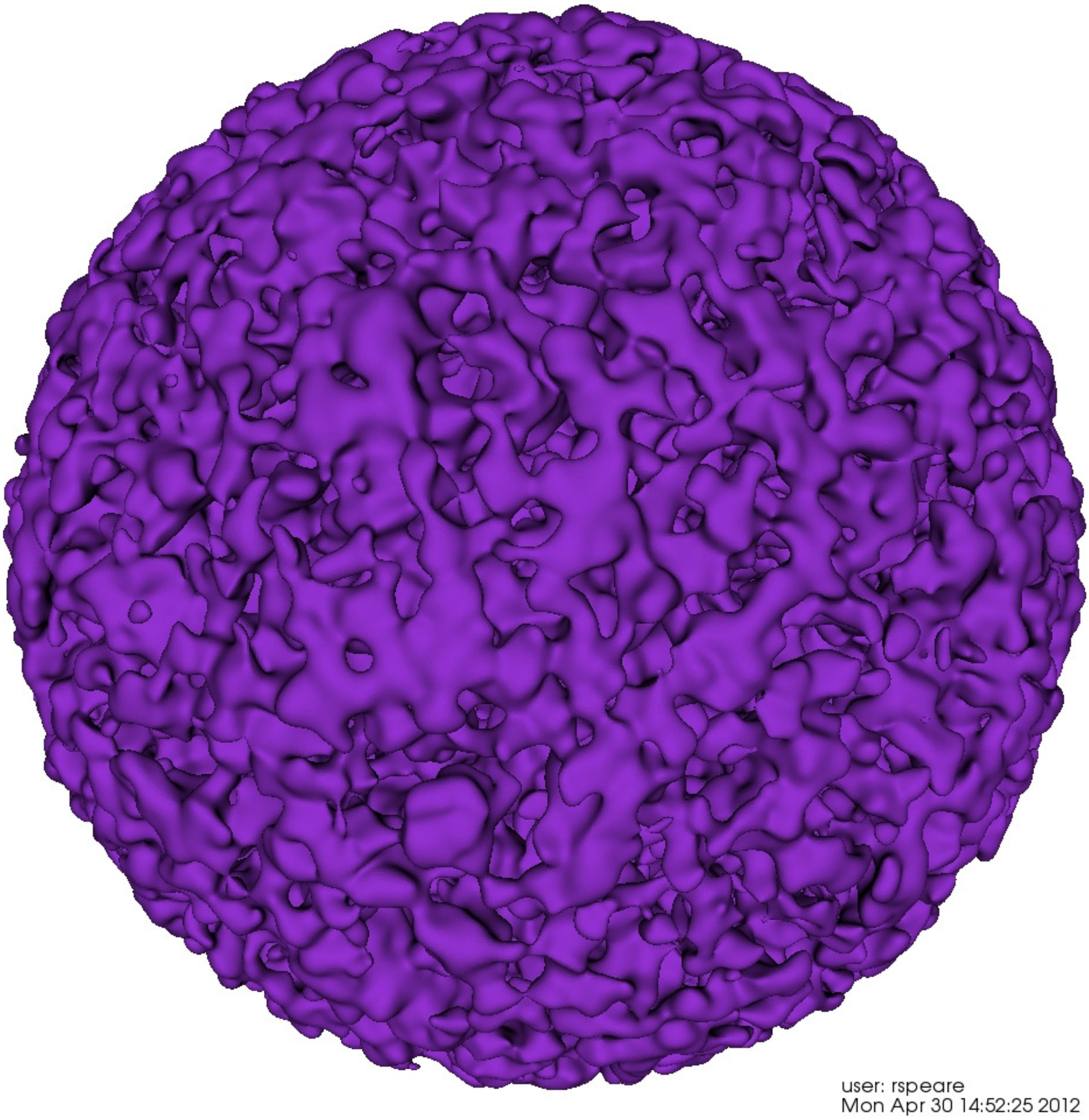

Figure 2: A smoothed 3-dimensional N-Body simulation. Note that before smoothing, this data would have looked very much like a dust plot, or a bunch of little stars – or sand particles – floating in space. Smoothing has smeared those tiny particles into a much larger, interesting structure.

Where, we have used the translation property above (Equation 47). Examining the obverse,

$$\int F(s)G(s'-s)dk = (F*G)(s'), \tag{63}$$

and transforming back to x-space,

$$\begin{aligned} \int\int F(s)G(s'-s)e^{2\pi i s' x}dsds' &= \overline{(F*G)(s')} && (64)\\ \int\int F(s)G(s'-s)e^{2\pi i s' x}ds'ds &= \overline{(F*G)(s')} && (65)\\ \int F(s)g(s)e^{2\pi i s x}ds &= \overline{(F*G)(s')} && (66)\\ f(x)g(x) &= \overline{(F*G)(s')} && (67) \end{aligned}$$

And this concludes the proof. □

1.8 Correlation

Another multiplication operation between square-integrable function is the correlation, represented by $f \star g$ and defined as:

$$(f \star g)(y) = \int f(x)g(x-y)dx \tag{68}$$

Notice how the final result is a function of y. x has once again been "integrated away". This correlation operation is basically taking two functions and displacing them by varying lengths, measuring the total area overlap. (Less complicated than convolution). We are essentially measuring the likeness between two distributions. Let f and g be real-valued functions, such that their transforms are complex Hermitian – $F^*(-k) = F(k)$. Let us examine this correlation operation in k-space

$$\begin{aligned} \overline{(f \star g)(y)} &= \int\int f(x)g(x-y)e^{-2\pi i s y}dxdy && (69)\\ &= \int\int f(x)g(x-y)e^{-2\pi i s y}dydx && (70)\\ &= \int f(x)G(-s)e^{-2\pi i s x}dx && (71)\\ &= F(s)G(-s) && (72)\\ &= F(s)G^*(s) && (73) \end{aligned}$$

Where, in the last step we use the complex Hermitian quality of $G(s)$. So, the transform of the correlation between two functions is the multiplication of the transforms, just like convolution – except one of those functions must be a complex conjugate. This has very interesting consequences for auto-correlation, or, a function's correlation with itself . . .

$$\begin{aligned}
\overline{(f \star f)(y)} &= \int\int f(x)f(x-y)e^{-iky}dxdy & (74)\\
&= \int\int f(x)f(x-y)e^{-iky}dydx & (75)\\
&= \int f(x)F(-k)e^{-ikx}dx & (76)\\
&= F(k)F(-k) & (77)\\
&= |F(k)|^2 & (78)
\end{aligned}$$

Strange, we have arrived at the square modulus of the transformed function! This leads us to Parseval's theorem, which states that the integral of the squared transform is equal to the integral of the squared function – an equal energy theorem. Let's take a look at the expectation value of auto correlation when $y = 0$:

$$\begin{aligned}
\langle (f \star f)(y)\rangle &= \langle |F(k)|^2\rangle & (79)\\
\langle (f \star f)(0)\rangle &= \langle |F(k)|^2\rangle & (80)\\
\langle |f(x)|^2\rangle &= \langle |F(k)|^2\rangle & (81)\\
& & (82)
\end{aligned}$$

Pretty cool, right? This is also referred to as Rayleigh's theorem – that the area under the squared Power spectrum is equal to the area under the squared two point correlation function. [1]

2 Examples

2.1 The Delta function

For the following section, we will adopt a new definition of the Fourier transform, which is more intuitive to Quantum Mechanics but carries along with a pesky normalization factor; a factor which, we will ignore for most of our discussion. The Fourier transform pairs are defined as:

$$\begin{aligned}
\overline{f(x)} = F(k) &= \frac{1}{2\pi}\int f(x)e^{-ikx}dx & (83)\\
\overline{F(k)} = f(x) &= \int F(k)e^{ikx}dk & (84)
\end{aligned}$$

The delta function $\delta(x)$ is defined to be zero everywhere except at the origin, or where $x = 0$ – this gets more complicated of course in higher dimensions like E_3. In fact, what's frustrating about the dirac delta function is that its not even a well defined function. It has no clear value without an integral sign and a test function. Let me explain:

$$\int_{-\infty}^{\infty} \delta(x)dx = 1 \tag{85}$$

The integral of the delta function is one.

$$\int_{-\infty}^{\infty} f(x)\delta(x)dx = f(0) \tag{86}$$

The integral of the delta function with some included test function f spits out that test function's value at the origin. It is more proper to call $\delta(x)$ a distribution in Schwartz space, or, a generalized function.

[1] In fact the correlation function $f \star f$ and $F \star F$ are incredibly important in cosmology and w/r/t Gaussian Random fields.

2.1.1 Aside: Generalized Functions

What is a generalized function? It is simply a function that is well-behaved under the operations of differentiation, integration, and translation. In fact, a generalized function does not have to have a single representation such as $f(x) = x$; it can be constructed as the limit of a sequence of *another* function! For example, the dirac delta function can be constructed out of an infinite sequence of Gaussians:

$$\delta(x) = \lim_{a\to 0} \frac{1}{a\sqrt{\pi}} e^{-x^2/a^2} \tag{87}$$

This sequence is *very* well-behaved, since a Gaussian is infinitely differentiable – or, is an element of the class of infinitely differentiable functions, C^∞. This is the most powerful formulation of the dirac delta function – another option is to use thinner and thinner top-hat or sinc functions .

2.1.2 The Delta's Transform

Let's treat our test function f as the complex exponential e^{-ikx}, this means

$$\int_{-\infty}^{\infty} f(x)\delta(x)dx = f(0) \tag{88}$$

$$\int_{-\infty}^{\infty} e^{-ikx}\delta(x)dx = 1 \tag{89}$$

$$\tag{90}$$

Strange, it seems the Fourier transform of the dirac delta function is unity! This implies a mutual transform: $1 \supset \delta$. We have now formed a transform pair.

Let us take a look at the delta function, offset by some value a:

$$\int_{-\infty}^{\infty} f(x)\delta(x-a)dx = f(a) \tag{91}$$

$$\int_{-\infty}^{\infty} e^{-ikx}\delta(x)dx = e^{-ika} \tag{92}$$

$$\tag{93}$$

Ok, so we have now found that $\overline{\delta(x-a)} = e^{-ika}$. Let's take a look from the other side:

$$\int_{-\infty}^{\infty} F(k)\delta(k-k')dk = F(k') \tag{94}$$

$$\int_{-\infty}^{\infty} e^{ikx}\delta(k-k')dk = e^{ik'x} \tag{95}$$

which means $\overline{\delta(k-k')} = e^{ik'x}$. We can easily extrapolate this to the Fourier transform, without the delta function underneath the integral:

$$\int_{-\infty}^{\infty} e^{-ikx}e^{ik'x}dx = \overline{e^{ik'x}} \tag{96}$$

$$\int_{-\infty}^{\infty} e^{-i(k-k')x}dx = \overline{e^{ik'x}} \tag{97}$$

$$\delta(k-k') = \overline{e^{ik'x}} \tag{98}$$

Where, we have used the fact that complex exponential construct a well defined orthogonal basis in L^2 space – i.e. the integral is zero unless $k = k'$. We can do the same analysis for dk integration, and summarizing our results below:

$$
\begin{align}
\delta(x) &\supset 1 \tag{99}\\
1 &\supset \delta(k) \tag{100}\\
e^{ik'x} &\supset \delta(k-k') \tag{101}\\
\delta(x-x') &\supset e^{-ikx'} \tag{102}
\end{align}
$$

Notice that I have not included the normalization factor for the Fourier transform that involves x integration. If I do this, we have to essentially multiply all of our δ function by 2π yielding,

$$
\begin{align}
2\pi\delta(x) &\supset 1 \tag{103}\\
1 &\supset 2\pi\delta(k) \tag{104}\\
e^{ik'x} &\supset 2\pi\delta(k-k') \tag{105}\\
2\pi\delta(x-x') &\supset e^{-ikx'} \tag{106}
\end{align}
$$

Interesting, we have now defined Fourier transform pairs for the delta function, which wasn't actually a well-defined function to begin with! To be more rigorous about this point: we have actually taken the fourier transform of a limiting sequence of functions, by taking the following steps:

1. Define a sequence of functions $f_N(x)$ such that $\lim_{N\to\infty} f_N = f(x)$. Make sure that f is square-integrable and well-behaved under differentiation and translation.
2. Compute the Fourier transform of the sequence $\overline{f_N(x)} = F_N(k)$.
3. Define the $F(k)$ as the limit of the transformed sequence, such that $F(k) = \lim_{N\to\infty} F_N(k)$.

So essentially, to find the Fourier transform of the dirac delta function, we could actually examine the limiting series of Fourier-transformed Gaussians – which are also Gaussians.

2.2 Sine and Cosine

Since we can build $\sin$ and $\cos$ out of complex exponential functions, let us demonstrate the power of "building transforms out of other transforms":

$$
\begin{align}
\sin(k'x) &= \frac{e^{ik'x} - e^{-ik\prime x}}{2i} \tag{107}\\
\overline{\sin(k'x)} &= \frac{\delta(k-k') - \delta(k+k')}{2i} \tag{108}\\
& \tag{109}
\end{align}
$$

And now for the cosine function:

$$
\begin{align}
\cos(k'x) &= \frac{e^{ik'x} + e^{-ik\prime x}}{2i} \tag{110}\\
\overline{\cos(k'x)} &= \frac{\delta(k-k') + \delta(k+k')}{2} \tag{111}\\
& \tag{112}
\end{align}
$$

Note that, these two transforms are "discrete" in the sense that there are specific values for k in which they are defined. A general remark can be made here, in the sense that the Fourier transform of periodic functions will always be discrete, since the complex exponential essentially picks out all of the "parallel" oscillating components of the function f.

2.3 The signum function

The signum function, $\mathrm{sgn}(x)$, is defined as -1 for $x < 0$ and 1 for $x \geq 0$. It is essentially a step function. Let us examine it's fourier transform:

$$\overline{\mathrm{sgn}(x)} = \int_{-\infty}^{\infty} \mathrm{sgn}(x) e^{-ikx} dx \tag{113}$$

Since the signum function is odd, we know by symmetry that the even portions of the complex exponential $e^{-ikx} = \cos(kx) - i\sin(kx)$ will add to zero, therefore we can re-write this integral using only the sine term:

$$\begin{aligned}
\overline{\mathrm{sgn}(x)} &= -i \int_{-\infty}^{\infty} \mathrm{sgn}(x) \sin(kx) \frac{dx}{2\pi} && (114)\\
&= -2i \int_{0}^{\infty} \mathrm{sgn}(x) \sin(kx) \frac{dx}{2\pi} && (115)\\
&= -2i \int_{0}^{\infty} \sin(kx) \frac{dx}{2\pi} && (116)\\
&= -\frac{i}{\pi} - \frac{\cos(kx)}{k}|_0^\infty && (117)\\
&= \frac{i}{\pi} \frac{\cos(kx)}{k}|_0^\infty && (118)
\end{aligned}$$

Cosine will not go to a well-defined value at the upper limit, and so we are left without a Fourier transform under its strict definition. But, if we construct the signum function as a generalized function, or, the limit of a sequence of functions, we can follow our steps from before:

1. Let's re-define the signum function:

$$f_N(x) = \begin{cases} e^{-x/N}, & \text{if } x < 0 \\ -e^{x/N}, & x \geq 0 \end{cases} \tag{119}$$

   The limit of this sequence of functions clearly goes to $\mathrm{sgn}(x)$.

2. Fourier transforming these two functions:

$$\begin{aligned}
\overline{f_N(x)} &= \int_0^\infty e^{-x/N}e^{-ikx}\frac{dx}{2\pi} + \int_{-\infty}^0 -e^{x/N}e^{-ikx}\frac{dx}{2\pi} && (120)\\
&= \int_0^\infty e^{-(\frac{1}{N}+ik)x}\frac{dx}{2\pi} + \int_{-\infty}^0 -e^{(\frac{1}{N}-ik)x}\frac{dx}{2\pi} && (121)\\
&= \frac{e^{-(\frac{1}{N}+ik)x}}{-(\frac{1}{N}+ik)x}|_0^\infty + \frac{-e^{(\frac{1}{N}-ik)x}}{(\frac{1}{N}-ik)x}|_0^\infty && (122)\\
&= \frac{1}{2\pi}\frac{1}{\frac{1}{N}+ik} + \frac{1}{2\pi}\frac{-1}{\frac{1}{N}-ik} && (123)\\
&= \frac{1}{2\pi}\frac{1}{\frac{1}{N}+ik} + \frac{1}{2\pi}\frac{-1}{\frac{1}{N}-ik} && (124)\\
&= \frac{1}{2\pi}(\frac{1}{\frac{1}{N}+ik} - \frac{1}{\frac{1}{N}-ik}) && (125)
\end{aligned}$$

3. And now taking the limit of the sequence of those transformed functions we find a much simpler expression

$$\begin{aligned}
\lim_{N\to\infty}\overline{f_N(x)} &= \frac{1}{2\pi}\frac{2}{ik} && (126)\\
\overline{\mathrm{sgn}(x)} &= \frac{-i}{\pi k} && (127)
\end{aligned}$$

It is amazing what you can accomplish with sequences of functions. Seemingly difficult $f(x)$ functions, at least in the sense of the Fourier transform, can be molded into much more tractable expressions.

2.4 The Heavy-Side Step function

The heavy-side step function is simply

$$H(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & x \geq 0 \end{cases} \tag{128}$$

Or, we can write the Heavy-side step function in terms of the signum function:

$$H(x) = \mathrm{sgn}(x) + 1 \tag{129}$$

This implies the the Fourier transform of the Heavy-side step function is the addition of the two transforms. But because the Heavy-side function is only non-zero for $x \geq 0$, while the signum function was defined for all x, we need to divide the transform by 2

$$\begin{aligned}
\overline{H(x)} &= \overline{\mathrm{sgn}(x)} + \overline{1} && (130)\\
&= \frac{1}{2}[\delta(k) - \frac{i}{\pi k}] && (131)
\end{aligned}$$

Interesting . . .

2.4.1 Aside: Convolution with the Heavyside

Notice that convolution with the Heavyside step function yields the integral of a function. For example

$$\begin{aligned} H * f &= \int_{-\infty}^{\infty} H(x) f(y-x) dx && (132)\\ f * H &= = H * f && (133)\\ f * H &= \int_{-\infty}^{\infty} H(y-x) f(x) dx && (134)\\ f * H &= \int_{-\infty}^{y} f(x) dx && (135) \end{aligned}$$

The flipped Heavyside function kills all output when its argument, $x' - x$ is negative. Therefore we can see that convolution with the Heavyside is equivalent to integration.

2.5 The Top Hat function

The Fourier transform of the top hat function, $\Pi(x)$ defined as

$$\Pi(x) = \begin{cases} 1/2, & |x| = 1/2 \\ 1, & |x| < 1/2 \\ 0, & \text{otherwise} \end{cases} \quad (136)$$

If we transform this function, we find that all x values are killed off outside of $|x| < 1/2$, so we have

$$\begin{aligned} \overline{\Pi(x)} &= \frac{1}{2\pi} \int_{-1/2}^{1/2} e^{-ikx} dx && (137)\\ &= \frac{1}{2\pi} \frac{e^{-ikx}}{-ik} |_{-1/2}^{1/2} && (138)\\ &= \frac{1}{2\pi} \frac{e^{-i\frac{k}{2}} - e^{ik\frac{1}{2}}}{-ik} && (139)\\ &= \frac{1}{2\pi} \frac{-2i \sin(k/2)}{-ik} && (140)\\ &= \frac{1}{2\pi} \frac{\sin(k/2)}{k/2} && (141)\\ & && (142) \end{aligned}$$

Without a normalization scheme, or, under the transform

$$F(s) = \int f(x) e^{-2\pi i s x} dx, \quad (143)$$

We find

$$\Pi(x) \supset \frac{\sin(s)}{s}. \quad (144)$$

2.6 The exponentially damped function

The Fourier transform of the exponentially damped function

$$f(x) = e^{-a|x|} \tag{145}$$

Is easy to do for $L_2(E_1)$. Let us integrate this transform in closed form

$$\begin{aligned} \overline{f(x)} &= F(s) && (146) \\ \overline{f(x)} &= \int_{-\infty}^{\infty} y(x)e^{-2\pi isx}dx && (147) \\ &= \int_{-\infty}^{\infty} Ae^{-a|x|}e^{-2\pi isx}dx && (148) \end{aligned}$$

Let us look at the positive side of the integral, where $x > 0$:

$$\begin{aligned} \int_0^{\infty} Ae^{-ax}e^{-2\pi isx}dx &= \int_0^{\infty} Ae^{-x(a+2\pi is)}dx && (149) \\ &= -\frac{Ae^{-x(a+2\pi is)}}{a+2\pi is}|_0^{\infty} && (150) \\ &= \frac{A}{a+2\pi is} && (151) \end{aligned}$$

Now for the negative side:

$$\begin{aligned} \int_{-\infty}^{0} Ae^{-ax}e^{-2\pi isx}dx &= \int_{-\infty}^{0} Ae^{x(a-2\pi is)}dx && (152) \\ &= \frac{Ae^{-x(a-2\pi is)}}{a-2\pi is}|_{-\infty}^{0} && (153) \\ &= \frac{A}{a-2\pi is} && (154) \end{aligned}$$

Adding these two complex numbers together, we get a real number (since they are complex conjugates of of one another):

$$F(s) = \frac{2Aa}{a^2+4\pi^2s^2} \tag{155}$$

And that's our transform. This is often called the Poisson kernel, and is used in $L_1(E_n)$ Fourier analysis much like a δ or ϵ ball, to "squeeze" a misbehaved function f through the Fourier transform.

2.7 Gaussians

Let us adopt the definition of the Fourier transform without normalization constants, i.e.:

$$\begin{aligned} f(x) &= \int F(s)e^{2\pi isx}ds && (156) \\ F(s) &= \int f(x)e^{-2\pi isx}dx && (157) \end{aligned}$$

If our $f(x) = e^{-\pi x^2}$, we can examine it's derivatives in both x and k space:

$$\begin{aligned}
\frac{\partial f(x)}{\partial x} &= -2\pi x f(x) && (158)\\
\mathcal{F}(\frac{\partial f(x)}{\partial x}) &= 2\pi i s F(s) && (159)\\
\mathcal{F}(-2\pi x f(x)) &= \frac{1}{i}\frac{\partial F(s)}{\partial s} && (160)\\
\mathcal{F}(-2\pi x f(x)) &= \mathcal{F}(\frac{\partial f(x)}{\partial x}) && (161)\\
\frac{1}{i}\frac{\partial F(s)}{\partial s} &= \mathcal{F}(\frac{\partial f(x)}{\partial x}) && (162)\\
\frac{1}{i}\frac{\partial F(s)}{\partial s} &= 2\pi i s F(s) && (163)\\
\frac{\partial F(s)}{\partial s} &= -2\pi s F(s) && (164)\\
\frac{dF}{F} &= -2\pi s ds && (165)\\
\log(F) &= -\pi s^2 + C && (166)\\
F(s) &= C_0 e^{-\pi s^2} && (167)\\
F(s) &\sim e^{-\pi s^2} && (168)
\end{aligned}$$

So, we have recovered a Gaussian in k-space! C_0 is most often unity, but this depends upon the normalization of $f(x)$. Note that $f(x)$ has a variance or width of $\frac{\sqrt{\pi}}{2}$, and $F(s)$ has a variance or width of $\frac{1}{\sqrt{2\pi}}$. As one Gaussian gets thinner the other gets fatter! (And, vice versa).This is intimately related to the uncertainty principle in Quantum mechanics – which, messes around with quite a few Gaussian wave packets to describe particles – the use of Large arrays of antennae to image extremely small points on the sky, and the various interference patterns created by diffraction gratings of small and big slits. We could go on and on about this curious property of squeezing in x-space turning into stretching in k-space; the point is, we now have our Fourier transform pair,

$$e^{-\pi x^2} \supset e^{-\pi s^2} \quad (169)$$

subject to a normalization scheme.

Part II
Statistics

3 In the x-space

3.1 Probability Density and Moments

Let us first define a probability density function, $P(x)$: the likelihood that a particle – or a state – will be found between the observable x and $x + \delta x$. This probability density function must be normalized, in the sense that, if we sum up all of the probabilities for all possible outcomes, we get unity:

$$\int P(x)dx = 1. \tag{170}$$

Let us start with an easy example. Take a Gaussian probability density function, with an arbitrary normalization constant placed next to it:

$$P(x) = C_0 e^{-\frac{x^2}{2\sigma^2}}, \tag{171}$$

where σ is the variance of the distribution, or the "width" (more on this later). Let us integrate this function by squaring and placing in polar coordinates

$$\begin{aligned}
(\int P(x)dx)^2 &= \int P(x)dx \int P(y)dy \\
&= C_0^2 \int\int e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} dxdy \\
&= C_0^2 \int\int e^{-\frac{x^2+y^2}{2\sigma^2}} dxdy \\
&= C_0^2 \int_0^{2\pi}\int_0^{\infty} e^{-\frac{r^2}{2\sigma^2}} rdrd\theta \\
&= 2\pi C_0^2 \int_0^{\infty} e^{-\frac{r^2}{2\sigma^2}} rdr \\
&= 2\pi C_0^2 \int_0^{\infty} e^u - \sigma^2 du \\
&= 2\pi\sigma^2 C_0^2(-e^u|_0^{\infty}) \\
&= 2\pi\sigma^2 C_0^2(1)
\end{aligned} \tag{172}$$

If the integral of each probability density function $P(x)$ is 1, this requires

$$C_0 = \frac{1}{\sqrt{2\pi\sigma^2}} \tag{173}$$

This is the standard normalization of a Gaussian.

Now that we have normalized our probability density function, we can derive meaningful statistics from these functions. The first is called the average, which is simply the weighted sum of the observable x over the domain of the probability density function:

$$\langle x \rangle = \int xP(x)dx \tag{174}$$

This is called the "first moment". We can relate this to the second, third and higher moments in the following way

$$\begin{aligned}
\left\langle x^2 \right\rangle &= \int x^2 P(x) dx && (175)\\
\left\langle x^3 \right\rangle &= \int x^3 P(x) dx && (176)\\
\left\langle x^n \right\rangle &= \int x^n P(x) dx && (177)
\end{aligned}$$

Let us refer to these moments synonymously as the expectation value of x to the "$n^{\rm th}$" power, or the "$n^{\rm th}$" moment: m_n. Recall that the variance, or the sum of the squares of deviation from the mean is defined as

$$\begin{aligned}
\sigma^2 &= \left\langle (x - \langle x \rangle)^2 \right\rangle && (178)\\
&= \left\langle (x - m_1)^2 \right\rangle && (179)\\
&= \left\langle (x^2 - 2m_1 x + m_1^2) \right\rangle && (180)\\
&= \int P(x)(x^2 - 2m_1 x + m_1^2) dx && (181)\\
&= \int P(x) x^2 dx - 2m_1 \int x P(x) dx + m_1^2 \int P(x) dx && (182)\\
&= m_2 - 2m_1 m_1 + m_1^2(1) && (183)\\
&= m_2 - m_1^2 && (184)\\
& && (185)
\end{aligned}$$

Where, we have pulled all the m_1 first moment terms out of the integral expressions and simplified the final expression. This variance, or "width" of the distribution is actually called the second cumulant, c_2, but more on that later.

3.2 Cumulative Distribution Function

The cumulative distribution function is normally labeled by the greek letter Φ, and is the probability of observing "any outcome up to x". It's essentially an integral:

$$\Phi(x) = \int_{-\infty}^{x} P(x') dx' \qquad (186)$$

For a normal distribution – or a Gaussian – we have:

$$\Phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{x} e^{\frac{-(x')^2}{2\sigma^2}} dx' \qquad (187)$$

If we take the derivative with respect to x we get back our Probability density:

$$\frac{d\Phi(x)}{dx}|_{x=x'} = P(x') \qquad (188)$$

This representation allows us to write our moments in a more compact way:

$$\langle x^n \rangle = \int x^n d\Phi \qquad (189)$$

Where, $d\Phi$ is now what's called a Lebesque measure.[2]

3.2.1 Error function

Building upon this notion of a cumulative distribution function for a Gaussian, we can now introduce the error function, which is often an abstruse concept. One can think of it as an altered cumulative distribution function for the normal

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \tag{190}$$

where we have essentially set the variance $\sigma^2 = \frac{1}{2}$ and integrated away from $x = 0$. If we understand that

$$\Phi(0) = \int_{-\infty}^{0} P(x')dx' = 1/2 \tag{191}$$

for the normal distribution, we can write

$$\text{erf}(x) = 2(\Phi(x) - \frac{1}{2}) \tag{192}$$

The error function, if we re-scale it and evaluate at $\frac{x}{\sigma\sqrt{2}}$, gives us the probability of observing an outcome a distance x from the mean value of a normal distribution; which, we have assumed to be centered at $x = 0$ for this entire discussion.

3.3 Moment Generating Function

Let us define now, the moment generating function:

$$M(k) = \int e^{kx} P(x)dx \tag{193}$$

Expands the above expression, we see that the exponential term carries all powers of x, and therefore we can isolate those powers by taking derivative with respect to k:

$$\begin{aligned} M(k) &= \int e^{kx} P(x)dx & (194)\\ M(k) &= \int (1 + kx + \frac{(kx)^2}{2!} + \frac{(kx)^3}{3!} + \frac{(kx)^4}{4!} + \ldots)P(x)dx & (195)\\ \frac{\partial^n M}{\partial k^n}|_{k=0} &= \int x^n P(x)dx & (196)\\ \frac{\partial^n M}{\partial k^n}|_{k=0} &= m_n & (197) \end{aligned}$$

Thus, we call $M(k)$ the moment generating function, because it's k derivatives about $k = 0$ give us all the moments of our probability distribution.

Another way to derive this result is to frame the moment generating function in terms of the laplace transform of the Probability density function:

$$M'(k) = \int e^{-kx} P(x)dx \tag{198}$$

[2] We can assume this measure is finite, since the cumulative distribution function by its definition must go to 1 or 0 as x goes to ∞ or $-\infty$ respectively.

Which, leaves us with the above definition of the n moments, except with an added sign convention

$$(-1)^n \frac{\partial^n M'}{\partial k^n}|_{k=0} = m_n \tag{199}$$

4 In the k-space

4.1 The Characteristic Function

Now that we have defined the moment generating function $M(k)$, and the closely related Laplace transform of the probability density function $P(x)$, $M'(k)$, we can define the fourier transform of the probability density function:

$$\phi(k) = M'(ik) \tag{200}$$
$$\phi(k) = \int e^{-ikx} P(x) dx \tag{201}$$

Or, in more compact notation one can write

$$P(x) \supset \phi(k) \tag{202}$$
$$\overline{P(x)} = \phi(k). \tag{203}$$

This characteristic function, by virtue of the Fourier transform has some very interesting properties:

1. As $P(x)$ gets thinner, $\phi(x)$ gets both wider and diminished, since for $a > 1$

$$\overline{P(ax)} = \frac{1}{a}\phi(k/a) \tag{204}$$

2. $\phi(0) = m_0$, or, a properly normalized PDF will yield $\phi(0) = 1$. This is related in higher dimensions to the fact that the supremum of ϕ is less than or equal to the 1 norm of P:

$$||\overline{P}||_\infty \leq ||P||_1 \tag{205}$$
$$||\phi||_\infty \leq ||P||_1 \tag{206}$$

3. Since the probability density function $P(x) \leq 1, \forall x$ we know that $P(x)^2 \leq P(x)$, allowing us to use Fourier analysis for L^2 integrable functions, which introduces a few more properties:

$$\int P(x)^2 dx = \int \phi(k)\phi^\star(k) dk \tag{207}$$
$$\int |P(x)|^2 dx = \int |\phi(k)|^2 dk \tag{208}$$
$$||P||_2 = ||\phi||_2 \tag{209}$$

Often called Parseval's theorem. We also can assume that $P(x)$ is the inverse fourier transform of $\phi(k)$ – now that we are in L^2 space, a small but important point – allowing us to write

$$P(x) = \frac{1}{2\pi}\int \phi(k) e^{ikx} dk \tag{210}$$

4. Finally, we can now formulate the moments of our probability density function in terms of the k derivatives evaluated about zero of our characteristic function:

$$\begin{aligned}
\phi(k) &= \int e^{-ikx}P(x)dx && (211)\\
\frac{\partial^n \phi}{\partial k^n}|_{k=0} &= \int (-ix)^n P(x)dx && (212)\\
\frac{\partial^n \phi}{\partial k^n}|_{k=0} &= (-i)^n m_n && (213)\\
i^n\frac{\partial^n \phi}{\partial k^n}|_{k=0} &= m_n && (214)
\end{aligned}$$

Pretty nifty right? Although, we have done nothing illuminating, this simply illustrates that properties of differentiating a fourier transform pair.

This characteristic function can be useful in the following way, let us use Eq. (210) and place $\phi(k)$ into the exponential term:

$$\begin{aligned}
P(x) &= \frac{1}{2\pi}\int e^{\log\phi(k)+ikx}dk && (215)\\
&= \frac{1}{\pi}\int e^{\psi(k)+ikx}dk && (216)
\end{aligned}$$

We can now examine what are called the “cumulants” of the distribution, which are created by the k derivative of our “cumulant generating function” $\psi(k)$ about zero. We define

$$i^n\frac{\partial^n \psi(k)}{\partial k^n}|_{k=0} = c_n \qquad (217)$$

Let’s examine the first few cumulants.

$$\begin{aligned}
\psi(k) &= \log\phi(k) && (218)\\
\frac{\partial \psi(k)}{\partial k}|_{k=0} &= \frac{1}{\phi(k)}\frac{\partial \phi}{\partial k}|_{k=0} && (219)\\
\frac{\partial \psi(k)}{\partial k}|_{k=0} &= \frac{1}{\phi(k)}\frac{\partial \phi}{\partial k}|_{k=0} && (220)\\
\frac{\partial \psi(k)}{\partial k}|_{k=0} &= \frac{m_1}{i} && (221)\\
& && (222)
\end{aligned}$$

This is the first cumulant: $c_1 = m_1$.

$$\begin{aligned}
\frac{\partial^2 \psi(k)}{\partial k^2}|_{k=0} &= \frac{-1}{\phi(k)^2}(\frac{\partial \phi}{\partial k})^2|_{k=0} + \frac{1}{\phi(k)}\frac{\partial^2 \phi}{\partial k^2}|_{k=0} && (223)\\
&= -(\frac{m_1}{i})^2 + \frac{m_2}{i^2} && (224)\\
i^2\frac{\partial^2 \psi(k)}{\partial k^2}|_{k=0} &= m_2 - m_1^2 && (225)\\
c_2 &= m_2 - m_1^2 && (226)
\end{aligned}$$

This is the second cumulant c_2, which is strangley enough, the variance of our probability density function. For a few more cumulants (we can leave these to the reader as an exercise):

$$\begin{aligned} c_1 &= m_1 && (227)\\ c_2 &= m_2 - m_1^2 && (228)\\ c_3 &= m_3 - 3m_2m_1 + 2m_1^2 && (229)\\ c_4 &= m_4 - 4m_3m_1 - 3m_2^2 + 12m_2m_1^2 - 6m_1^4 && (230) \end{aligned}$$

We can now write our cumulant generating function in a taylor expansion

$$\psi(k) = -ic_1k - \frac{k^2}{2}c_2 + i\frac{k^3}{3!}c_3 + \frac{k^4}{4!}c_4 + \ldots \tag{231}$$

Notice that $\psi(0) = 0$. This expansion will be very useful to us later.

4.2 Addition of random Variables: Addition of Cumulants

Let us begin with a theorem in statistics, which leads to some very interesting properties in terms of the cumulants and adding two random independent variables:

Theorem 2. *Let X and Y be two independent random variables with density functions $f_x(x)$ and $f_y(y)$, then the sum $z = x + y$ is a random variable with density function $f_z(z)$, where f_z is the convolution of f_x and f_y.*

Proof. Let us construct the cumulative density function for the variable z, which is a sum of two independent random variables $z = x + y$,

$$\begin{aligned} F(x+y) &= \int\int P(x)P(y)dxdy && (232)\\ F(x+y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{z-y} P(x)P(y)dxdy && (233)\\ F(x+y) &= \int_{-\infty}^{\infty} F(z-y')P(y')dy' && (234)\\ & && (235) \end{aligned}$$

Taking the derivative of this cumulative density function F we get the convolution of our two density functions:

$$\begin{aligned} \frac{\partial F(x+y)}{\partial z} &= \int_{-\infty}^{\infty} \frac{\partial F(z-y')}{\partial z} P(y')dy' && (236)\\ P(x+y) &= \int_{-\infty}^{\infty} \{\frac{\partial}{\partial z}\int_{-\infty}^{z-y} P(x)dx\} P(y')dy' && (237)\\ P_z &= \int_{-\infty}^{\infty} P(z-y')P(y')dy' && (238)\\ P_z &= (P_x * P_y)(z) \qquad \square && (239) \end{aligned}$$

And so we find that the new probability density function is the convolution of the two former density functions.

So, if we want to describe the sum of two independent variables, we can represent the resulting characteristic function ϕ_z as the multiplication of the two initial characteristic functions. These leads to a curious property of the cumulants:

$$\begin{aligned}
\phi(k_z) &= \phi(k_x)\phi(k_y) && (240)\\
P(z) &= \int \phi(k_x)\phi(k_y)e^{ik_z z}dk && (241)\\
&= \int e^{\log\phi(k_x)+\log\phi(k_y)}e^{ik_z z}dk && (242)\\
&= \int e^{\psi(k_x)+\psi(k_y)}e^{ik_z z}dk && (243)\\
\psi(k_z) &= \psi(k_x)+\psi(k_y) && (244)\\
i^n\frac{\partial^n\psi(k_z)}{\partial k^n} &= c_{xn}+c_{yn} && (245)\\
c_{zn} &= c_{xn}+c_{yn} && (246)
\end{aligned}$$

Thus, cumulants are additive under convolution. For example, if a random variable X has a density function f_x with variance σ_x^2, we know that by Theorem (2), adding another independent random variable Y with density function f_y and variance σ_y^2, we have a new probability density function f_z with variance $\sigma_z^2 = \sigma_x^2 + \sigma_y^2$.

This concept is incredibly important for random walks, where one has a “blind” turtle that takes random steps within euclidean space. The final position of the turtle can be viewed as a random variable Z which is the sum of all the independent but identical steps $Z = \Sigma_i X_i$. The variance of this distribution will of course be the sum of the variance's of each and every step along the way, and so will therefore be proportional to the square root of the number of steps, $\sigma_z \sim \sqrt{N}$. We will also find the characteristic function of Z, $\phi(k_z)$ will be the multiplication of all the former steps' characteristic function $\phi(k_x)$, which leads immediately to the central limit theorem.

4.2.1 Addition of Random Variables 2: A Note on ”Support”

Now, note that in the former discussion, we expected the two random variables to be in the range $-\infty \to \infty$. This range of possible values is called the support for a probability density function. This worked out quite well above, in that we had a standard convolution to combine probability densities. But what if the support of our random variables X, Y is, say, $0 \to \infty$?

Let's take a look, using the same methods

$$\begin{aligned}
S &= X+Y && (247)\\
X &\sim f(x) && (248)\\
Y &\sim g(y) && (249)\\
S &\sim\ ? && (250)
\end{aligned}$$

We can now write $X = S - Y$, but we can't integrate our variable y from zero to infinity, because that would imply X has a negative value!!! The limiting case of our random variable Y in this setup is S, corresponding to $X = 0$. And so we have:

$$\begin{aligned}\Phi(S) &= P(S \leq s) = \int_0^s F(s-y)g(y)dy && (251)\\ \frac{\partial \Phi(S)}{\partial s} = P(s) &= \int_0^s f(s-y)g(y)dy && (252)\\ & && (253)\end{aligned}$$

So, now our integration range has fundamentally changed, but no matter, we see the corrollary. We are essentially performing a ”convolution” or overlap of the probability densities, but over our proper support $0 \to s$.

4.2.2 Multiplication of two Random Variables

For the multiplication of two random variables on the support $0 \to \infty$, let us start with the setup

$$\begin{aligned} Z &= XY && (254)\\ X &\sim f && (255)\\ Y &\sim g && (256)\\ Z &\sim ? && (257)\end{aligned}$$

We write

$$\begin{aligned}\Phi(z) = P(Z \leq z) &= \int_0^\infty F(\frac{z}{y})g(y)dy && (258)\\ P(z) &= \int_0^\infty f(\frac{z}{y})g(y)\frac{dy}{y} && (259)\end{aligned}$$

4.2.3 Square of a Random Variable

Now let

$$\begin{aligned} Z &= X^2 && (260)\\ X &\sim f && (261)\\ Z = X^2 &\sim ? && (262)\end{aligned}$$

We write

$$\begin{aligned}\Phi(z) = P(Z \leq z) &= P(x \leq \sqrt{z}) - P(x \leq -\sqrt{z}) && (263)\\ &= F(\sqrt{z}) - F(-\sqrt{z}) && (264)\\ P(z) &= \frac{1}{2\sqrt{z}}\left[f(\sqrt{z}) + f(-\sqrt{z})\right] && (265)\\ P(x^2) &= \frac{1}{2x}\left[f(x) + f(-x)\right] && (266)\end{aligned}$$

If $f(x) = f(-x)$ – is an even function – we can write:

$$P(x^2) = \frac{f(x)}{x} \qquad (267)$$

4.2.4 Square Root of a Random Variable

Note that, if we define Z as the **square root** of a random variable, we have to treat the situation differently:

$$\begin{aligned} Z &= \sqrt{X} && (268) \\ X &\sim f && (269) \\ Z \sim & \; ? && (270) \end{aligned}$$

We can write

$$\begin{aligned} \Phi(z) &= P(Z \leq z) = \Phi(X \leq z^2) && (271) \\ \frac{\partial \Phi(z)}{\partial z} = g(z) &= 2zf(z^2) && (272) \\ g(\sqrt{x}) &= 2\sqrt{x}f(x) && (273) \end{aligned}$$

So we do not have a clean inverse of moving between squares and square roots of random variables; there is this pesky factor of two lying around here!

4.3 Sample Spaces and Treating PDF's as Waves

For two observables x and y, we associate two sample sample spaces, which consist of all the possible ways of measuring outcome x. Specifically we can write the probability of observing outcome x as the ratio N_x – the number of outcomes which constitute x being measured – and N – the total number of outcomes in the sample space.

$$P(x) = \frac{N_x}{N} \qquad (274)$$

We can even construct concepts such as conditional probability based on the intersection and union of sample spaces.

4.4 The Multidimensional Central Limit Theorem

Let us begin with a brief review; the definition of a normalized probability distribution:

$$\int_{-\infty}^{\infty} P(x)dx = 1 \qquad (275)$$

and a moment,

$$\int_{-\infty}^{\infty} x^n P(x)dx = m_n \qquad (276)$$

Notice that m_1 is the 'mean' or average, or, 'center of mass' of the Probability distribution. The second moment, m_2 is intimately related to the variance, or width of our distribution $P(x)$.

We can define the moment generating function as

$$\int e^{kx} P(x)dx = M(k) \qquad (277)$$

Where now, taylor expanding our exponential integrand, we see that the n^{th} derivative of $M(k)$ with respect to k evaluated at $k = 0$, gives the respective moments:

$$M(k) = \int e^{kx} P(x) dx \tag{278}$$

$$= \int \Sigma_n \frac{(kx)^n}{n!} P(x) dx \tag{279}$$

$$\frac{\partial^n M}{\partial k^n}|_{k=0} = \int x^n P(x) dx \tag{280}$$

$$\frac{\partial^n M}{\partial k^n}|_{k=0} = m_n \tag{281}$$

We can also define the moment generating function as the laplace transform of $P(x)$, which leaves us with an added sign convention in the definition of our moments,

$$M'(k) = \int e^{-kx} P(x) dx \tag{282}$$

$$(-1)^n \frac{\partial^n M'}{\partial k^n}|_{k=0} = m_n \tag{283}$$

Now we are in a position to define the fourier transorm of our probability distribution, called the characteristic function,

$$M'(ik) = \phi(k) = \int e^{-ikx} P(x) dx \tag{284}$$

We can simply think of $\phi(k)$ as the fourier transform of $P(x)$, and note that $\phi(0) = 1$ always – as required by our normalization condition above.

Writing the inverse fourier transform, we can now write $P(x)$ in terms of $k-$space components,

$$P(x) = \frac{1}{2\pi} \int e^{ikx} \phi(k) dk \tag{285}$$

$$= \frac{1}{\pi} \int e^{ikx} e^{\log(\phi(k))} dk \tag{286}$$

Where I have used the $\frac{1}{2\pi}$ for inverse fourier normalization conventions (not the same as the QM convention!).

I would now like to define cumulants which, are the corresponding derivatives of the $\psi(k) = \log(\phi(k))$ function, seen the exponential argument of the above equation. First, notice that under the fourier transform, multiplication in x-space is differentiation in k-space, i.e.

$$\mathcal{F}[xP(x)] = (-i)\frac{\partial \phi}{\partial k} \tag{287}$$

and so we find that the various moments can be defined by the partial derivatives of the characteristic function (not much different than what we did before with the moment generating function M', under simply a change of variable):

$$m_n = (-i)^n \frac{\partial^n \phi}{\partial k^n}|_{k=0} \tag{288}$$

Using our $\psi(k)$ function from before, we can now define the cumulants

$$c_n = (i)^n \frac{\partial^n \psi}{\partial k^n}|_{k=0} \tag{289}$$

which, can be constructed out of our initial moments,

$$\begin{aligned}
c_1 &= m_1 && (290)\\
c_2 &= m_2 - m_1^2 && (291)\\
c_3 &= m_2 - 3m_2m_1 + 2m_1^2 && (292)\\
c_4 &= m_4 - 4m_3m_1 - 3m_2^2 + 12m_2m_1^2 - 6m_1^2 && (293)
\end{aligned}$$

Notice that the first cumulant is our standard mean, or average, and that the second cumulant is our definition of variance, or the expectation value of difference from the mean. (i.e. $c_1 = \bar{x}$ and $c_2 = \sigma^2 = E[(x - \bar{x})^2]$). These cumulants are very good descriptors of a statistical distribution because they add under convolution. Let's take a closer look.

Under the fourier transform, convolution in x-space is multiplication in k-space – see convolution theorem – and so cumulants and cumulant-generating functions add under convolution. In probability theory, when one adds two statistically independent variables x and y in order to create a new variable $z = x+y$, then the probability distribution that describes z is the convolution of the two former probability distributions: $P(z) = (P_x \star P_y)(z)$. Let's see what such an addition would do in fourier space,

$$\begin{aligned}
P(z) &= (P_x \star P_y)(z) && (294)\\
&= \mathcal{F}^{-1}\left[\phi_x(k)\phi_y(k)\right] && (295)\\
&= \frac{1}{2\pi}\int e^{ikx}\phi_x(k)\phi_y(k)dk && (296)\\
&= \frac{1}{2\pi}\int e^{ikx}e^{\log(\phi_x(k))}e^{\log(\phi_y(k))}dk && (297)\\
&= \frac{1}{2\pi}\int e^{ikx}e^{\psi_x(k)+\psi_y(k)}dk && (298)\\
& && (299)
\end{aligned}$$

Expanding both cumulant-generating functions as a taylor series, and collecting like powers of k, we find

$$\begin{aligned}
P(z) &= \frac{1}{2\pi}\int e^{ikx}e^{-ic_{1x}k-c_{2x}k^2/2+ic_{3x}\frac{k^3}{3!}+\dots}e^{-ic_{1y}k-c_{2y}k^2/2+ic_{3y}\frac{k^3}{3!}+\dots}dk && (300)\\
&= \frac{1}{2\pi}\int e^{ikx}e^{-i(c_{1x}+c_{1y})k-(c_{2x}+c_{2y})k^2/2+i(c_{3x}+c_{3y})\frac{k^3}{3!}+\dots}dk && (301)\\
& && (302)
\end{aligned}$$

Notice that $c_{1z} = c_{1x} + c_{1y}$ and $c_{2z} = c_{2x} + c_{2y}$, or the mean is the sum of the two former means, and the variance is the sum of the two former variances. Very cool!

Under what are called identically independent processes, such as random walks, we have a new variable $z = \sigma x_i$, which is a superposition of independent variables that are described by the exact same probability density function. In this case, our expectation value – or first moment – for the variable z becomes $Nm_{1z} = N\bar{x}$, where N is the number of 'steps' or 'trials' taken. Similarly we find that the variance of the sum of random variables is $\sigma_z^2 = N\sigma_x^2$. Or the variance grows as $\sqrt{N}$.

The central limit theorem depends intimately upon this addition of cumulants under convolution, and uses the inverse properties of expansion and dilation under the fourier transform to truncate the taylor expansion of our effective cumulant generating functions. Let's take a look at an 'iid' process, or $P(z)$, where $z = \Sigma x_i$.

$$\begin{aligned}
P(z) &= \frac{1}{2\pi}\int e^{ikz}\Pi_{i=1}^{N}\phi_i(k)dk && (303)\\
P(z) &= \frac{1}{2\pi}\int e^{ikz}e^{\Sigma_i\psi_i(k)}dk && (304)\\
P(z) &= \frac{1}{2\pi}\int e^{ikz}e^{\Sigma_i(-ic_{1i})-ik}e^{\Sigma_i(-c_{2i})\frac{-k^2}{2}} + \cdots dk && (305)
\end{aligned}$$

As we convolve more and more probability density functions, our variances will add, and we will end up with an extremely wide density function. In fourier space, this corresponds to an extremely narrow characteristic function, and allows us to assume negligible value of $\phi(k)$ at high k-values. Thus, we can write

$$\psi(k) = -ic_1k - c_2\frac{k^2}{2} + ic_3\frac{k^3}{3!} - c_4\frac{k^4}{4!} + \ldots \qquad (306)$$

as

$$\psi(k) \approx -ic_1k - c_2\frac{k^2}{2} \qquad (307)$$

Taking this into account – and noting that the c_1 and c_2 are in fact sums of former means and variances under convolution – we can now write $P(z)$ as,

$$\begin{aligned}
P(z) &= \frac{1}{2\pi}\int e^{ikz}e^{-ic_1k}e^{-c_2\frac{k^2}{2}}dk && (308)\\
P(z) &= \frac{1}{2\pi}\int e^{ikz}e^{-ic_1k}e^{-c_2\frac{k^2}{2}}dk && (309)
\end{aligned}$$

Setting $z' = z - c_1$, or, centering about the mean we can exclude the first exponential and write

$$P(z') = \frac{1}{2\pi}\int e^{ikz'}e^{-c_2\frac{k^2}{2}}dk \qquad (310)$$

This is the fourier transform of a gaussian, which is itself a gaussian,

$$\begin{aligned}
P(z') &= \frac{1}{\sqrt{2\pi c_2}}e^{\frac{-z^2}{2c_2}} && (311)\\
P(z) &= \frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(z-c_1)^2}{2\sigma^2}} && (312)\\
P(z) &= \frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(z-\bar{z})^2}{2\sigma^2}} && (313)
\end{aligned}$$

Whew! Notice that our mean and variance in this final equation are simply the sums of former means and variances of the independent variables, x_i that created z. ($z = \Sigma_i x_i$; $c_1 = \Sigma_i c_{1i}$; $c_2 = \Sigma_i c_{2i}$). The central limit theorem does not depend on each of these variables – perhaps N of them – being identical, or, described by the same probability density function, they need only be independent.

Now, extrapolation to multiple dimension is not to bad, we now deal with random vectors as compared to random scalars. Let's represent these vectors with boldface, and label their components with a superscript. We can now write

$$\mathbf{z} \;=\; \Sigma_j \mathbf{x}^{\mathbf{j}} \tag{314}$$

Where the superscript denotes the j^{th} independent random vector. Similarly, our k space is now multidimensional, and so we write the forward fourier transform of our probability density as

$$\phi(\mathbf{k}) \;=\; \int e^{i\mathbf{k}\cdot\mathbf{x}} P(\mathbf{x}) d^d\mathbf{x} \tag{315}$$

$$\phi(\mathbf{k}) \;=\; \int e^{i\mathbf{k_i}\mathbf{x_i}} P(\mathbf{x}) d^d\mathbf{x} \tag{316}$$

The cumulants and the moments are now rank n tensors, seen by the following

$$\frac{\partial^n M}{\partial \mathbf{k}_i \partial \mathbf{k}_j \cdots \partial \mathbf{k}_\gamma}|_{\mathbf{k}=0} \;=\; (-i)^n \mathbf{m}_{ij\cdots\gamma} \tag{317}$$

$$\frac{\partial^n \psi}{\partial \mathbf{k}_i \partial \mathbf{k}_j \cdots \partial \mathbf{k}_\gamma}|_{\mathbf{k}=0} \;=\; (-i)^n \mathbf{c}_{ij\cdots\gamma} \tag{318}$$

We now use the same approximation scheme as before, convolving an absurd number of multi-dimensional probability density functions, $P(\mathbf{x}_i)$ in order to yield a convolution in fourier space – and thus an addition of cumulant generating functions

$$P(\mathbf{z}) \;=\; \frac{1}{(2\pi)^d} \int e^{i\mathbf{k}\cdot\mathbf{z}} e^{\Sigma\psi_i(\mathbf{k})} d^d\mathbf{k} \tag{319}$$

$$P(\mathbf{z}) \;=\; \frac{1}{(2\pi)^d} \int e^{i\mathbf{k}\cdot\mathbf{z}} e^{-i\mathbf{c}_i\mathbf{k}_i)} e^{-i\mathbf{k}_i\mathbf{c}_{ij}\mathbf{k}_j)} e^{-i\mathbf{c}_{ijk}\mathbf{k}_i\mathbf{k}_j\mathbf{k}_k)} \cdots d^d\mathbf{k} \tag{320}$$

$$\tag{321}$$

Truncating our taylor expansion and centering about the multidimensional mean $\mathbf{z}'_i = \mathbf{z}_i - \mathbf{c}_i$,

$$P(\mathbf{z}') \;=\; \frac{1}{(2\pi)^d} \int e^{i\mathbf{k}\cdot\mathbf{z}'} e^{-i\mathbf{k}_i\mathbf{c}_{ij}\mathbf{k}_j)} d^d\mathbf{k} \tag{322}$$

$$\tag{323}$$

We can rescale coordinates by writing

$$\mathbf{w} \;=\; \sqrt{\mathbf{c}_{ij}}\mathbf{k}_j \tag{324}$$

where, the square of a matrix can be written terms of its diagonalization by two unitary matrices $\mathbf{S}_{ij}$, and the diagonal matrix of eigenvalues $\mathbf{\Lambda}_{ij} = \delta_{ij}\lambda_i$,

$$\mathbf{c}_{ij} = \mathbf{S}^{-1}\mathbf{\Lambda}\mathbf{S} \tag{325}$$

$$\mathbf{c}_{ij} = \mathbf{S}_{il}\lambda_l\delta_{lm}\mathbf{S}_{lj} \tag{326}$$

$$\sqrt{\mathbf{c}_{ij}} = \mathbf{S}_{il}\sqrt{\lambda_l}\delta_{lm}\mathbf{S}_{lj} \tag{327}$$

$$\mathbf{w} = \sqrt{\mathbf{c}_{ij}}\mathbf{k}_j \tag{328}$$

The determinant of this matrix will be the product of the eigenvalues λ_j, which is also the dilating factor by which one expands a single $\mathbf{k}$ vector; so we can now write our infinitesimal volume element in d-dimensional $\mathbf{k}$ space using this determinant,

$$d^d\mathbf{w} = (|\mathbf{c}_{ij}|)^{d/2}\, d^d\mathbf{k} \tag{329}$$

Now rewriting our integral, we have a multi-dimensional transform of a Gaussian, which is another Gaussian

$$P(\mathbf{z}') = \frac{1}{(2\pi)^d}\int e^{i\frac{\mathbf{w}_j}{\sqrt{\mathbf{c_{ij}}}}\mathbf{z}'_i} e^{-\frac{\mathbf{w}_i\mathbf{w}_i}{2}} \frac{d^d\mathbf{w}}{|\mathbf{c}_{ij}|^{d/2}} \tag{330}$$

$$P(\mathbf{z}') = \frac{1}{(2\pi|\mathbf{c}_{ij}|)^{d/2}}\exp\left(-\frac{1}{2}\frac{\mathbf{z}'_l}{\mathbf{c}_{il}}\frac{\mathbf{z}'_\gamma}{\mathbf{c}_{i\gamma}}\right) \tag{331}$$

Whew! The argument in the exponential – namely, those nasty matrix multiplications into the second cumulant matrices – can be simplified, yielding a single covariance matrix on the bottom.

$$P(\mathbf{z}') = \frac{1}{(2\pi|\mathbf{c}_{ij}|)^{d/2}}\exp\left(-\frac{1}{2}\frac{\mathbf{z}'_l}{\mathbf{c}_{il}}\frac{\mathbf{z}'_\gamma}{\mathbf{c}_{i\gamma}}\right) \tag{332}$$

5 Connection with Linear Stochastic ODE's

So a few friends of mine are working on Stochastic ODE's and their connection to path integrals. After dorking out about this for a few moments, I'm able to make some "baby" statements about the problem. If you consider a sequence of random numbers:

$$\{\mathbf{X}_i\}_{i=1}^n \tag{333}$$

which is determined by the following difference equation:

$$d\mathbf{X}_i = \mathbf{X}_{i+1} - \mathbf{X}_i = a_i + \mathbf{W}_i \tag{334}$$

subject to the initial condition $\mathbf{X}_0 = 0$

You can express the solution as a sum of two sums – one deterministic and one random.

$$X_n = \sum_{i=0}^n a_i + \sum_{i=1}^n \mathbf{W_i} \tag{335}$$

Where I have boldfaced all random variables. For instance a_i is a real sequence of numbers, perhaps they are the same for all i. $\mathbf{W}$ is a noise variable, or some random forcing function. We see that the solution after N steps will be

$$\mathbf{X}_n = na + \sum_{i=1}^{n} \mathbf{W}_i \tag{336}$$

Now, if we see that $\mathbf{W}_i$ is drawn from some probability distribution at every single step i, we know that, at asymptotic times $N \to \infty$, subject to certain conditions on the probability density of W_i, our distribution on $\mathbf{X}$ will converge to a Gaussian. This is very cool, and not necessarily dependent on $\mathbf{W}$ being an identically independently distributed variable. We simply say that if

$$\mathbf{W_i} \sim N(0,\sigma^2)\ \forall i \tag{337}$$

then,

$$\mathbf{X_N} \sim na(t) + N(0, n\sigma^2) \tag{338}$$

Where $N(0,\sigma^2)$ stands for a normal distribution with zero mean and variance σ^2. Note that, this is simply a conclusion from the addition of cumulants under convolution – which is what you do when add random variables.

$$Z = X + Y \tag{339}$$

$$X \sim N(c_1, c_2) \tag{340}$$

$$Y \sim N(c_1', c_2') \tag{341}$$

$$Z \sim N(c_1 + c_1', c_2 + c_2') \tag{342}$$

So our cumulants add, and the central limit theorem hinges upon this, because since our characteristic function – or the fourier transform of our probability distribution – is bounded above by one (1), when we convolve tow distributions in real space we multiply in frequency space, making the characteristic function of our result variable Z – which is very much like an average, thinner and thinner and thinner... Meaning that you can truncate the characteristic function's cumulant generating function ψ at order k^2, leading to a Gaussian.

This means that any sum of random variables, even they are not identically and independently distributed – although they must be independent in order to convolve – and even if those variables have non-zero higher order cumulants, like skewness c_3 or kurtosis c_4, will give you a Gaussian in the $n \to \infty$ limit. This is an analog of the law of large numbers.

So why do we care in this Stochastic ODE case? It means that under linear dynamics, at asymptotic times, we converge to a Gaussian distribution on $\mathbf{X}$, even our noise function itself has very strange properties, like higher order cumulants. This is very strange indeed, and comes from the fact that system is **linear**, i.e. we are **adding** random variables together.

Under non-linear evolution, it can be shown using Perturbation theory that non-zero third and higher order moments are created, but showing this in the stochastic framework is a bit difficult ...

It is easy to show however, that an equation like:

$$L_0 \delta = \delta^2 \tag{343}$$

where L_0 is some linear differential operator, can be expanded in power series of small parameter λ

$$L_0 \delta = \lambda \delta^2 \tag{344}$$

$$\delta = \sum_{i=1}^{\infty} \lambda^i \delta_i \tag{345}$$

So we have, to each order:

$$\lambda^0 : L_0\delta_0 \;=\; 0 \tag{346}$$

which is our linear solution. Then we have to leading order:

$$\lambda^1 : L_0\delta_1 \;=\; \delta_0^2 \tag{347}$$

Now we find, that if we take the connected third moment, or the third cumulant, we get a nonzero value:

$$\langle\delta^3\rangle \;=\; \langle\delta_0^3\rangle + \lambda\langle\delta_0^2\delta_1\rangle + \ldots \tag{348}$$

If δ_0 is Gaussian distributed, as we found that we would be for some driving function at asymptotic times – or if we simply assume Gaussian initial conditions – then we know that $\langle\delta_0^3\rangle = 0$. The leading order term however, will not be zero, because it goes like $\sim\delta_0^4$, which under Wick's theorem/Gaussian statistics can be built out of second moments.

So we see that "Gaussian" nonlinearity will drive one away from Gaussianity. But the question is, how to express this in stochastic differential equations, which seem to show that even for very strange "noise" functions, our asymptotic solutions for $\mathbf{X}$ go Gaussian.

6 The Gram Charlier Expansion

For distributions that are "mildly non-gaussian" there is a clever expansion of

7 A few standard Distributions

7.1 The Binomial Disribution

The binomial distribution is a description of indistinguishable successes and failures, given a number of trials. For example, let us ask how many steps forward a drunken man is likely to take if he has a probability p of stepping forward – success – and a probability q of "staggering" in place – failure. Let each step be of unit length. Let the drunken man make a total of N "stepping trials".

If each and every step is distinguishable, we can label them as

$$s_1, s_2, s_3, \ldots s_N, \tag{349}$$

and there are $N!$ ways of ordering these steps; since one has N choices of placing s_1, then $(N-1)$ choices of placing s_2, and so on. This is the case where successful trials are distinguishable. But we are only interested in the final number of forward steps – or, the final number of successes – and so these steps are indistinguishable. Therefore, there are

$$\left(\begin{array}{c} N \\ k \end{array}\right) = \frac{N!}{(N-k)!k!}$$

. ways of taking k steps forward. We say this because the $k!$ orderings of successes are indistinguishable, and the $(N-k)!$ orderings of failures are indistinguishable. For every success we must multiply our total probability by p, and for every failure we must multiply by q, and so we find the Probability of taking k steps forward is:

$$P(k) \;=\; \left(\begin{array}{c} N \\ k \end{array}\right) p^k q^{N-k}. \tag{350}$$

This is called the binomial distribution.

7.1.1 Binomial Distribution in the Limit

If we imagine our drunken man taking many, many, many steps, or $N \to \infty$, then the expression above can be simplified using some curious identities. Let us first establish $q = 1 - p$, and note that as N gets very large, we can write our Binomial distribution as:

$$P(k) = \frac{N!}{k!(N-k)!} p^k q^{N-k}. \tag{351}$$

Let's take a quick look at those large factorials, using something call **stirling's approximation**

7.2 Stirling's Approximation and the Poisson Distribution

To take a quick detour, let us examine the following definition of the factorial:

$$N! = \int_0^\infty x^N e^{-x} dx \tag{352}$$

One way to prove this is to write

$$I(a) \int_0^\infty e^{-ax} dx = \frac{1}{a} \tag{353}$$

and take the derivative underneath the integral sign, to write:

$$\begin{aligned} I'(a) &= \frac{\partial}{\partial a} \int_0^\infty e^{-ax} dx && (354) \\ &= \int_0^\infty -axe^{-ax} dx && (355) \\ &= \frac{-1}{a^2} && (356) \end{aligned}$$

and more generally,

$$\frac{\partial^n I(a)}{\partial a^n} = (-1)^n \int_0^\infty a^n x^n e^{-ax} dx = \frac{(-1)^n n!}{a^{n+1}} \tag{357}$$

Setting $a = 1$ we find

$$\Gamma[n+1] = \int_0^\infty x^n e^{-x} dx = n! \tag{358}$$

Now let's examine this integral in the limit $n \to \infty$. We can take our x argument up, into the exponential and write the corresponding function as $f(x)$:

$$\begin{aligned} n! &= \int_0^\infty e^{-x + n\log x} dx && (359) \\ &= \int_0^\infty e^{f(x)} dx && (360) \\ f(x) &= -x + n\log x && (361) \end{aligned}$$

Now $f(x)$ is an absurdly large – or high-valued – function for large n, and so we can approximate this integral as only "counting" contributions around the maximum of $f(x)$. We find the position this maximum in the normal way:

$$\begin{aligned} f' &= -1 + \frac{n}{x} = 0 && (362) \\ x_0 &= n && (363) \end{aligned}$$

Taking a look at our second derivative

$$f''|_{x_0} = -\frac{n}{x^2} = -\frac{1}{n} < 0 \qquad (364)$$

we see that x_0 is the position of a true maximum. Expanding out our $f(x)$ with a Taylor approximation:

$$\begin{aligned} n! &\approx \int_0^\infty e^{f(x_0)+f'(x)|_{x_0}(x-x_0)+f''(x)|_{x_0}\frac{(x-x_0)^2}{2}}dx && (365) \\ & && (366) \end{aligned}$$

We see that the first derivative term is zero by construction, and we are left with a constant times a Gaussian,

$$\begin{aligned} n! &\approx e^{-n+n\log n}\int_0^\infty e^{\frac{(x-x_0)^2}{2n}}dx && (367) \\ &\approx n^n e^{-n}\int_0^\infty e^{\frac{(x-n)^2}{2n}}dx && (368) \\ & && (369) \end{aligned}$$

Now this integral is tricky, because we are taking the limit as $n \to \infty$, which means that, essentially, the middle of our Gaussian distribution is far afield from $x = 0$. Since the integral of any Gaussian $e^{-x^2/(2\sigma^2)}$ is $\sqrt{2\pi\sigma^2}$, we can approximate the integral above to be the "full" $-\infty < x < \infty$ integration, because our moment, or center of the distribution, x_0 is far to the positive side of zero. This yields, with $\sigma^2 = \sqrt{n}$:

$$n! \approx n^n e^{-n}\sqrt{2\pi n} \qquad (370)$$

Which is the so-called Stirling's approximation.

Now if we use this approximation to examine the binomial distribution in the same limit:

$$P(k;N) = \frac{N!}{k!(N-k)!}p^k(1-p)^{N-k} \qquad (371)$$

We write our factorials as:

$$P(k;N) = \frac{1}{k!}\frac{\sqrt{2\pi N}N^N e^{-N}}{\sqrt{2\pi(N-k)}(N-k)^{(N-k)}e^{-N+k}}p^k(1-p)^{N-k} \qquad (372)$$

Cancelling our $\sqrt{2\pi}$ terms and writing $\lambda = Np$, the expected value of our coin flips, given N trials and proabability p:

$$\begin{aligned}
P(k;N) &\approx \frac{1}{k!}\frac{\sqrt{N}N^Ne^{-N}}{\sqrt{(N-k)}(N-k)^{(N-k)}e^{-N+k}}\left(\frac{\lambda}{N}\right)^k\left(1-\frac{\lambda}{N}\right)^{N-k} && (373)\\
& \ldots \text{cancelling our square roots} \ldots && (374)\\
&\approx \frac{1}{k!}\frac{N^Ne^{-N}}{N^{N-k}e^{-N+k}}\left(\frac{\lambda}{N}\right)^k\left(1-\frac{\lambda}{N}\right)^{N}\left(1-\frac{\lambda}{N}\right)^{-k} && (375)\\
&\approx \frac{1}{k!}\frac{N^Ne^{-N}}{N^{N-k}e^{-N+k}}\left(\frac{\lambda}{N}\right)^k e^{-\lambda}\left(1-\frac{\lambda}{N}\right)^{-k} && (376)\\
&\approx \frac{\lambda^k}{k!}\frac{N^Ne^{-N}}{N^Ne^{-N+k}}e^{-\lambda}\left(1-\frac{\lambda}{N}\right)^{-k} && (377)\\
&\approx \frac{\lambda^ke^{-\lambda}}{k!}\frac{1}{e^k}\left(1-\frac{\lambda}{N}\right)^{-k} && (378)\\
&\approx \frac{\lambda^ke^{-\lambda}}{k!} && (379)
\end{aligned}$$

So we find, that in the large $N \to \infty$ limit, our binomial distribution becomes a poisson distribution, characterized by an "infinite" number of samples N – or coin flips – and an expectation value λ.

7.2.1 Properties of the Poisson

7.2.2 The Exponentially Damped PDF

Now, another way to get to the Poisson distribution is through something called an exponential distribution. Let X be a random variable between zero and infinity, described by the probability density g.

$$\begin{aligned}
X &\sim g(x) && (380)\\
0 < X &< \infty && (381)\\
g(x) &= \alpha e^{-\alpha x} && (382)
\end{aligned}$$

$g(x)$ is a "memoryless" probability distribution, in the sense that, if we construct the cumulative probability function,

$$\begin{aligned}
\Phi(x) &= = P(X \leq x) = \int_0^x g(x)dx = 1 - e^{-\alpha x} && (383)\\
P(X \geq x) = e^{-\alpha x} & && (384)
\end{aligned}$$

We see that

$$P(X \geq y+z) = P(Y \geq y)P(Z \geq z) \qquad (385)$$

This simple rule of multiplication means that events are not conditionally related. Observe that, for this distribution, the mean is $m_1 = 1/\alpha$ and second cumulant $c_2 = 1/\alpha^2$.

Now imagine we were adding a number of "memoryless", Random variables $X_1, X_2, X_3, \ldots$, each determined by same the exponential distribution $g(x)$. This would be similar to asking, "What's the Probability of the total waiting time for N line-queue-ers to be s?"

$$\begin{aligned}
s &= X_1 + X_2 + \ldots X_N && (386)\\
X_i &\sim g && (387)
\end{aligned}$$

We have learned that adding Random variables can be described as convolving the probability densities, and so we find

$$s \sim g_N = (g \star g \star g \cdots \star g)(s) \text{ “N convolutions”} \tag{388}$$

We can find this probability density with a recursion relation:

$$g_{n+1}(s) = \int_0^\infty g_n(s-x)g(x)dx \tag{389}$$

Looking at our base case:

$$\begin{aligned} g_2(s) &= \int_0^s \alpha^2 e^{-\alpha(s-x)} e^{-\alpha x} dx && (390) \\ &= \int_0^s \alpha^2 e^{-\alpha s} dx && (391) \\ &= \alpha^2 s e^{-\alpha s} && (392) \end{aligned}$$

Continuing on, we will find that each sucessive integration brings in another factor of α and performs an incomplete Gamma integration:

$$\begin{aligned} g_3(s) &= \int_0^s \alpha^3 (s-x) e^{-\alpha s} dx && (393) \\ &= \alpha e^{-\alpha x} \gamma(3, \frac{s-x}{\alpha}) && (394) \\ \gamma(n,a) &= \int_0^a x^{n-1} e^{-x} dx && (395) \end{aligned}$$

This more generally results in

$$g_n(s) = \alpha \frac{(\alpha s)^{n-1}}{(n-1)!} e^{-\alpha s} \tag{396}$$

Or, the probability that the total waiting time ”S” – the sum of waiting times X of n identical processes – is between $s + \delta s$ and $s - \delta s$.

Now if we are to construct the conditional probability of this distribution, that will be a very tricky process, since we can see that the integral would involve doing many, many integrations by parts. We can do this, nonetheless by tabulation, and write:

$$\begin{aligned} G_{n+1}(s) = \int_0^s g_{n+s}(s)ds &= 1 - e^{-\alpha s}\left(1 + \alpha s + \frac{(\alpha s)^2}{2} + \frac{(\alpha s)^3}{3!} \cdots + \frac{(\alpha s)^n}{n!}\right) && (397) \\ &= 1 - e^{-\alpha s}\left(e^{\alpha s} - \sum_{i=n+1}^{\infty} \frac{(\alpha s)^i}{i!}\right) && (398) \\ &= e^{-\alpha s} \sum_{i=n+1}^{\infty} \frac{(\alpha s)^i}{i!} && (399) \\ & && (400) \end{aligned}$$

This is the probability of exactly $n+1$ processes with random variable waiting times adding up to S or less than S. Or written more clearly:

$$X_i \sim g(x) \tag{401}$$

$$G_{n+1}(s) = P(X_1 + X_2 + \cdots \leq s) \tag{402}$$

Now if we want the probability that **exactly** n processes add up to the waiting total time s, we could take the difference

$$\begin{aligned} G_{n+1}(s) - G_n(s) &= e^{-\alpha s} \sum_{i=n+1}^{\infty} \frac{(\alpha s)^i}{i!} - e^{-\alpha s} \sum_{i=n}^{\infty} \frac{(\alpha s)^i}{i!} && (403) \\ &= \frac{(\alpha s)^n}{n!} e^{-\alpha s} && (404) \end{aligned}$$

Strange, because s is our total observation time, and α is the expected value for a single process. This looks a lot like our former expected value for an *entire* sampling period N in the case of coin flips for $N \to \infty$, just a more continuous case.

Writing $\alpha s = \lambda$, we get back the Poisson distribution (!):

$$G_{k+1}(s) - G_k(s) = P(n; \lambda) = \frac{\lambda^k}{k!} e^{-\lambda} \tag{405}$$

7.3 Gamma Densities

Now, we have already constructed our exponential "memoryless" distributions $g(x)$ and $g_n(s)$, convolving the probability densities with each other in order to describe the sum of waiting times. We can also construct something called a Gamma Probability density, which is like $g_n(s)$, but accounts for having "half" or non-integer members n:

$$f_{\alpha,\nu}(x) = \frac{1}{\Gamma(\nu)} \alpha^{\nu} x^{\nu-1} e^{-\alpha x} \tag{406}$$

For integer values of ν we get back our former density $g_n(s)$.

These Gamma densities have a nice property in that they are closed under convolutions:

$$(f_{\alpha,\mu} \star f_{\alpha,\nu})(x) = f_{\alpha,\mu+\nu}(x) \tag{407}$$

And also, we note that the mean and variance of these Gamma densities are:

$$m_1 = \frac{\nu}{\alpha} = \nu \alpha^{-1} \tag{408}$$

$$c_2 = \frac{\nu}{\alpha^2} = \nu \alpha^{-2} \tag{409}$$

Or simply, ν times the typical mean and variance of a single exponential distribution. The gamma densities, for integer ν can be used to model the waiting time of ν simultaneous processes.

7.4 The Chi-Squared Distribution

Let us now talk about the sum of the squares of a random variable, where each one is determined by a Gaussian distribution:

$$X \sim \eta(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}} \tag{410}$$

This normal distribution can actually be written in terms of a Gamma density:

$$\frac{\eta(x)}{x} = f_{\frac{1}{2\sigma^2},\frac{1}{2}}(x^2) \tag{411}$$

And so we see that, if we were to construct a random variable Z, which is the square of a random variable X, we would have

$$Z = X^2 \tag{412}$$

$$X \sim \eta(x) \tag{413}$$

$$X^2 \sim \frac{\eta(x)}{x} = f_{\frac{1}{2\sigma^2},\frac{1}{2}}(x^2) \tag{414}$$

If we are to add up multiple Z_i variables, such that we have, for instance,

$$\chi^2 = \sum_i^d Z_i \tag{415}$$

$$\chi^2 = \sum_i^d X_i^2 \tag{416}$$

$$\tag{417}$$

Then χ^2 is described by a simple convolution of the probability densities – the gamma $f_{\frac{1}{2\sigma^2},\frac{1}{2}}$ – given above!

$$\chi^2 \sim f_{\frac{1}{2\sigma^2},\frac{d}{2}}(\chi^2) \tag{418}$$

$$f(\chi^2) = \frac{\chi^{d-2}}{\Gamma\left(\frac{d}{2}\right)} \frac{1}{2^{d/2}\sigma^N} e^{\frac{-\chi^2}{2\sigma^2}} \tag{419}$$

$$g(\chi) = 2\sqrt{\chi^2} f(\chi^2) = \frac{\chi^{d-1}}{\Gamma\left(\frac{d}{2}\right)} \frac{1}{2^{(d/2-1)}\sigma^N} e^{\frac{-\chi^2}{2\sigma^2}} \tag{420}$$

This final probability distribution could be described as the radial density ($\chi \to r$) of a random walker in d-dimensions, where each step is governed by a gaussian distribution of variance σ. (If you want to look ahead to more advanced things, take a look at the volume of a d-dimensional ball (Section 29.4) and see how the normalization factors compare).

Now, let us relabel the variances σ^2 as the effective variance after N steps $\sigma^2 = N\sigma_1^2$, for the $N = 2, 3$ case, we have:

$$P_2(R) = \frac{R}{N\sigma^2} \mathrm{Exp}\left[\frac{-\mathrm{R}^2}{2\mathrm{N}\sigma^2}\right] \tag{421}$$

$$P_3(R) = \frac{R^2}{\sqrt{\frac{\pi}{2}}(N\sigma)^{3/2}} \mathrm{Exp}\left[\frac{-\mathrm{R}^2}{2\mathrm{N}\sigma^2}\right] \tag{422}$$

The two dimensional case – without the N's – is the standard **Rayleigh Distribution**.

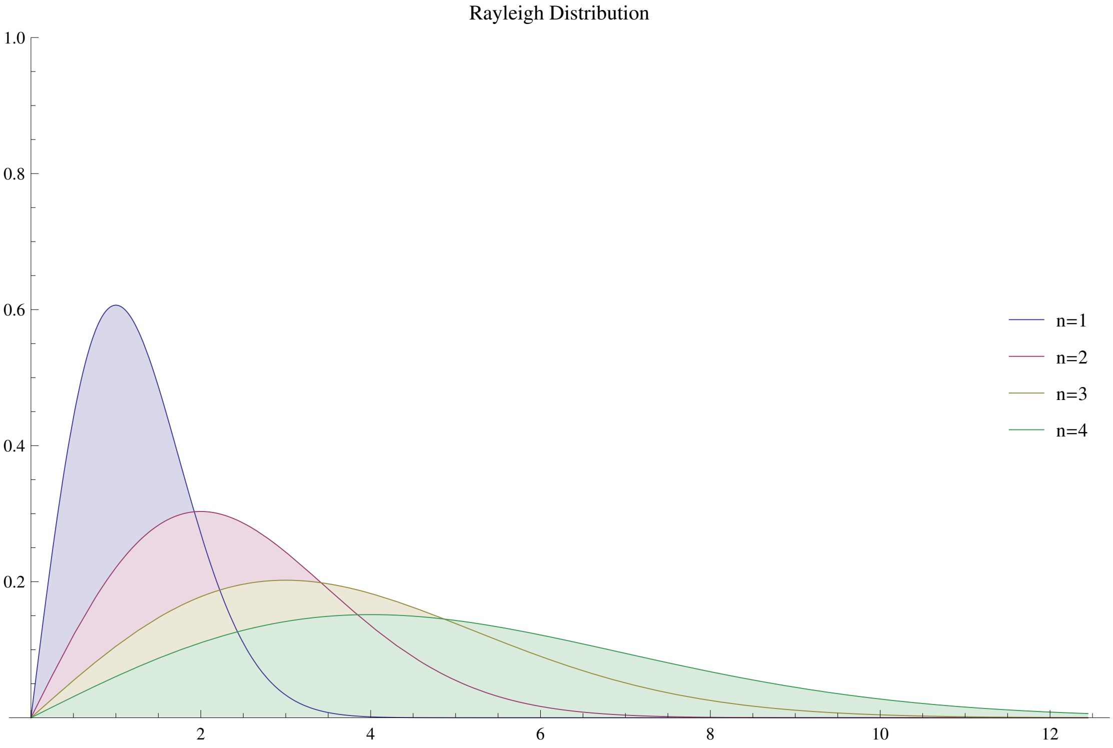

Figure 3: Rayleigh Distribution for 2-dimensions, $f(x) = \frac{x}{n}e^{\frac{-x^2}{2n}}$ with unit variance $\sigma = 1$ and the number of steps n equal to various integers. One can see immediately that, for a random walker in two dimensions, the number of steps has a profound effect on the probability density of radius. Our expectation for r is "smeared" out with more and more steps, n.

8 In the Sample space

In probability theory, we use sets to describe something called a sample space, which is a collection of all the possible outcomes of an experiment. Say we are interested in the probability of an outcome A. We represent this probability with the symbol $P(A)$, and its calculation is a simple ratio:

$$P(A) = \frac{\text{\# of Sample Space members that yield outcome A}}{\text{Total \# of sample space members}} \tag{423}$$

Note that with this equation, we can have multiple members that yield the same example. For example, say we flip a quarter two times. Our sample space, S, is:

$$\begin{gathered} TT \\ TH \\ HT \\ HH \end{gathered}$$

Four members. If we are interested in the probability of flipping one heads and one tails, we find $P(1H) = \frac{2}{4}$, because there are two members of the sample space that yield this outcome. The probability of flipping two heads is $P(2H) = \frac{1}{4}$, because there is only one satisfactory member of the sample space. Conversely, for two tails, we find $P(2T) = \frac{1}{4}$. If we add up all of these probabilities, we find:

$$P(2T) + P(2H) + P(1H) = 1$$

Which is required for any proper normalized probability density – except now, we're using discrete counting of sample space members to determine probability.

8.0.1 N Flips of a coin

The case gets a bit harder if we flip a coin N times. Now, the sample space will contain 2^N members, and if we are interested in the probability of flipping 2 "heads", then we must use combinatorics to describe our probability. If we take into account that Tails and Heads flips are indistinguishable, then there are

$$\left(\begin{array}{c} n \\ 2 \end{array}\right) = \frac{n!}{2!(n-2)!}$$

members of the sample space which yield a total of two heads! If we compare this to the binomial distribution, where p is the probability of a "success" – heads – and q is the probability of a "failure" – tails, then the probability of two flips is a simple binomial distribution:

$$P(2\text{ heads}) = \left(\begin{array}{c} n \\ 2 \end{array}\right) p^2 q^{N-2}$$

If we note that, for a flipping coin $p = q = \frac{1}{2}$, we find,

$$\begin{aligned} P(2\text{ heads}) &= \left(\begin{array}{c} n \\ 2 \end{array}\right) (\frac{1}{2})^2 (\frac{1}{2})^{N-2} \\ P(2\text{heads}) &= \left(\begin{array}{c} n \\ 2 \end{array}\right) \frac{1}{2}^N \\ P(2\text{ heads}) &= \frac{\left(\begin{array}{c} n \\ 2 \end{array}\right)}{2^N} \end{aligned}$$

And this final equation is simply a ratio of the number of sample space members that yield two heads, divided by the total number of sample space members. Pretty cool right? Our probability of an outcome of k heads is simply the ratio

$$P(k \text{ heads}) = \frac{\begin{pmatrix} n \\ k \end{pmatrix}}{2^N} \tag{424}$$

.

Which, is intimately related to the following identity:

$$2^n = 1 + \begin{pmatrix} n \\ 1 \end{pmatrix} + \begin{pmatrix} n \\ 2 \end{pmatrix} + \begin{pmatrix} n \\ 3 \end{pmatrix} + \dots \begin{pmatrix} n \\ n-1 \end{pmatrix} + \begin{pmatrix} n \\ n \end{pmatrix} = \Omega \tag{425}$$

We see that this equation is a representation of our sample space for n coin flips! This sample space has Ω equally probable members. In statistical mechanics Ω represents the number of ”accessible” microstates, or the number of ways in which a system can orient itself ”quantum mechanically” (since, after all, small systems must make discrete & definite choices of things like spin, angular momentum, etc. just like the flipping of a coin.)

8.0.2 Conditional Probability

When dealing with multiple outcomes, say A and B we are often concerned with the likelihood of outcome A given B has already happened. We normally write this as

$$P(A|B). \tag{426}$$

And conversely, the probability of outcome B, given A has already happened is

$$P(B|A). \tag{427}$$

These are tricky concepts to deal with, because we are not sure if

$$P(B|A) = P(A), \tag{428}$$

which means outcome B has no effect upon the outcome A – the two events are said to be uncorrelated, or, statistically independent. One can represent this conditional probability with an equation, called Bayes' theorem, which states:

$$P(B)P(A|B) = P(A \cap B) \tag{429}$$

In words the right hand side of the equation reads: ”The probability that outcome A and B both happen.” The left hand-side of the equation reads ”the probability that B happens time the probability that A happens, given B has already happened. Summarizing: ”The probability that A and B happen, is equal to the probability that B happens, times the probability that A happens, given B has already happens”. Enough casuistry. What the heck does that mean? This is intimately related to a tree diagram, where on represents branching probabilities to reach final results.

Exercise: Create such a conditional probability tree for three coin flips, drawing out the diagram for the four possible results TT, TH, HT, HH. What is the conditional probability of the second flip? Are two coin flips statistically independent? See Fig. 8.0.2 for solution.

One can now see how conditional probabilities are simply the “second branches” in a probability tree.

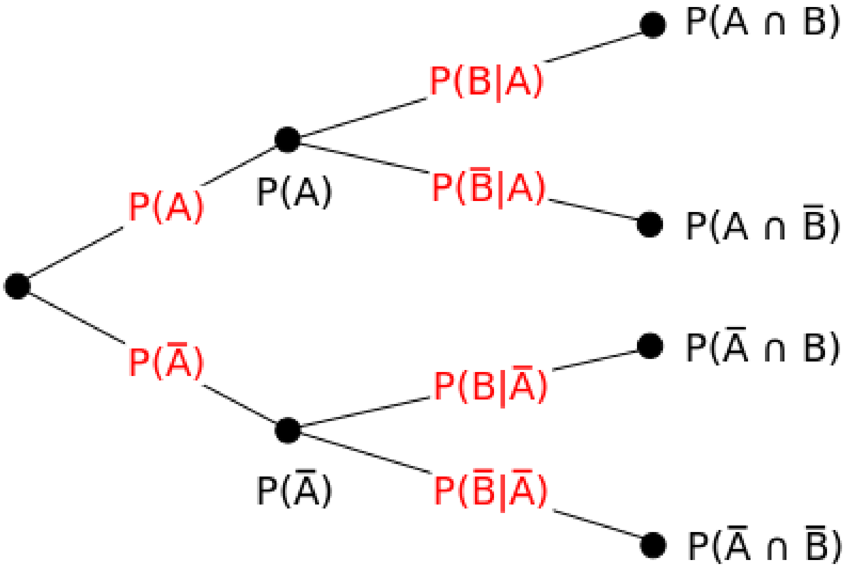

Figure 4: Notice how our first branch is split between outcome A happening, $P(A)$, and outcome A not happening $P(\overline{A})$. The branching probabilities after this first event are "conditional", in the sense that they depend upon prior results. The second branches lead to our four final outcomes.

8.0.3 Set Definitions and Identities

For a quick review of sets, let us represent the outcomes A and B with a Venn-Diagram. We create two circles, one which contains all the possible ways of getting outcome A, another circle which contains all the possible outcomes of B. Both circles will be embedded in a rectangle, which represents the total sample space S.

With this Venn-Diagram, we can now say that:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \tag{430}$$

This will prove extremely useful when playing with conditional probability. Let us go back to our earlier expressions for conditional probability of the two outcomes A and B, except we will swap the order of observation:

$$P(A)P(B|A) = P(A \cap B) \tag{431}$$

$$P(B)P(A|B) = P(B \cap A) \tag{432}$$

Since the set intersection operator is commutative – $(A \cap B) = (B \cap A)$ – we can combine the two conditional probabilities above and write

$$P(x|y) = \frac{P(X)P(y|x)}{P(Y)} \tag{433}$$

$$P(x|y) = \frac{P(X \cap Y)}{P(Y)} \tag{434}$$

These are slightly more refined versions of Bayes' theorem.

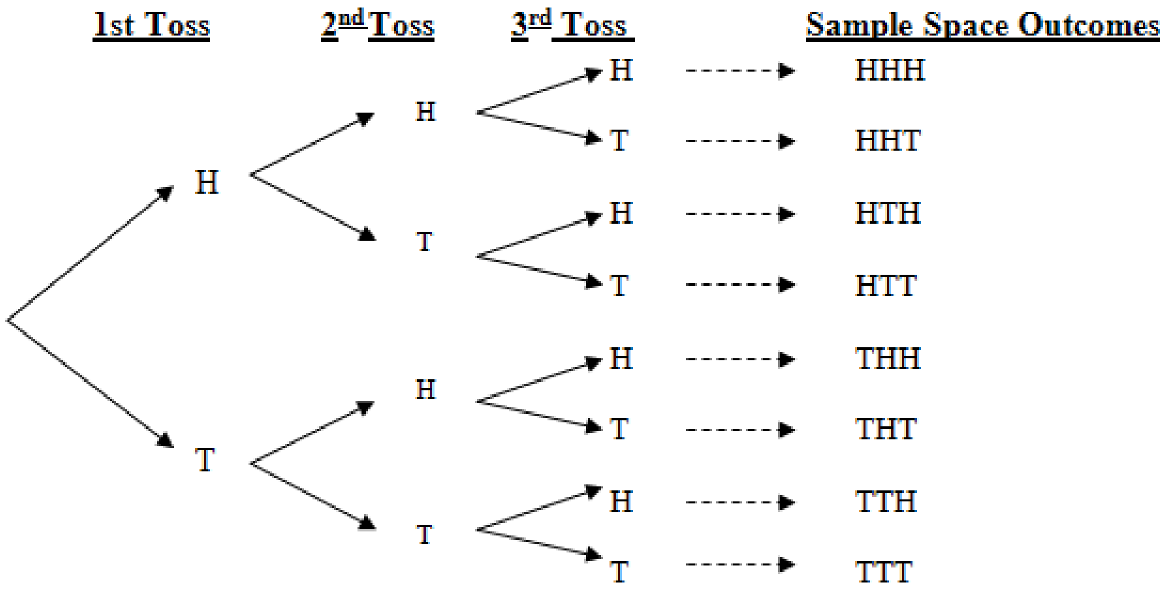

Figure 5: **Exercise Solution:** Notice how our sample space is composed of eight total outcomes. The probability of each branch is simply $\frac{1}{2}$, meaning each "flip trial" is statistically independent from previous flips.

8.0.4 Partitioning the Sample Space

Let us motivate this concept with an important example. One of the most stressful outcomes of medicine is a positive test for some rare disease. Let's take Cancer. Normally, lab tests are described by their accuracy, given a Cancer-ridden patient. For example:

$$P(+|C) = .99 \tag{435}$$

Or, given a cancer-ridden patien, the test will return positive, 99% of the time. Conversely, we also have an associated accuracy for negative tests, e.g.:

$$P(-|NC) = .95 \tag{436}$$

Or, given a non-cancer-ridden patient, the test will correctly return negative 95% of the time. We are interested in the following:

$$P(+|NC) = \text{Probability of a false Positive} \tag{437}$$

$$P(-|C) = \text{Probability of a false Negative} \tag{438}$$

Let's focus on the false positive case. If we make a simple chart, we can see that the probability of getting a positive test is described by:

$$P(+) = P(+|NC)P(NC) + P(+|C)P(C) \tag{439}$$

$$P(+) = P(+\cap NC) + P(+\cap C) \tag{440}$$

We were able to write these equations because the outcomes C and NC fill up the entire sample space S. In fact, for any set of events H_i whose sum fills up the sample space,

$$\sum_i H_i = S, \tag{441}$$

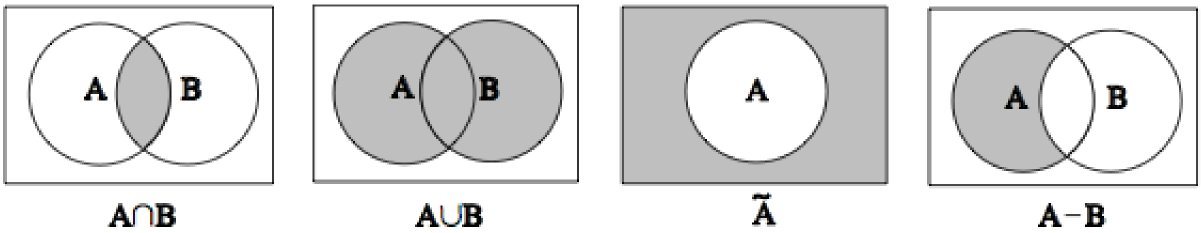

Figure 6: Notice how the intersection of A and B, or $A \cap B$ is represented by the middle region: it is the probability of outcome A *and* B. Notice how $A \cup B$ is the entire area of both circles – or the probability of outcome A *or* B. If outcomes A and B do not exhaust the sample space, then there is supposedly an outer region of possibilities – the surrounding rectangle. This diagram uses an overline $\overline{A}$ to represent outcomes *other* than A.

we can write

$$P(A) = \sum_i P(A|H_i)P(H_i). \tag{442}$$

which means that Bayes' equation can be written:

$$P(A|B) = \frac{P(A \cap B)}{\sum_i P(B|H_i)P(H_i)} \tag{443}$$

Or, if B is one of the H_i, we can write:

$$\begin{aligned} P(A|H_j) &= \frac{P(A \cap H_j)}{\sum_i P(B|H_i)P(H_i)} && (444) \\ &= \frac{P(H_j|A)P(H_j)}{\sum_i P(B|H_i)P(H_i)} && (445) \end{aligned}$$

Wow, that's a big equation! But for our purposes, in this equation, we have $H_i = C, NC$. Therefore, we can write the probability of a false positive being:

$$P(NC|+) = \frac{P(+|NC)P(NC)}{P(+|NC)P(NC) + P(+|C)P(C)} \tag{446}$$

Let's play with this equation, and put everything in terms of $P(NC)$. First we replace $P(C)$ with $1 - P(NC)$.

$$P(NC|+) = \frac{P(+|NC)P(NC)}{P(+|NC)P(NC)) + P(+|C)(1 - P(NC))} \tag{447}$$

Next, we substitute $P(+|NC) = 1 - P(+|C)$,

$$= \frac{(1 - P(+|C))P(NC)}{(1 - P(+|C))P(NC)) + P(+|C)(1 - P(NC))} \tag{448}$$

Notice how this result only depends upon the accuracy of the positive test: $P(+|C)$. Let's call this variable A – for accuracy.

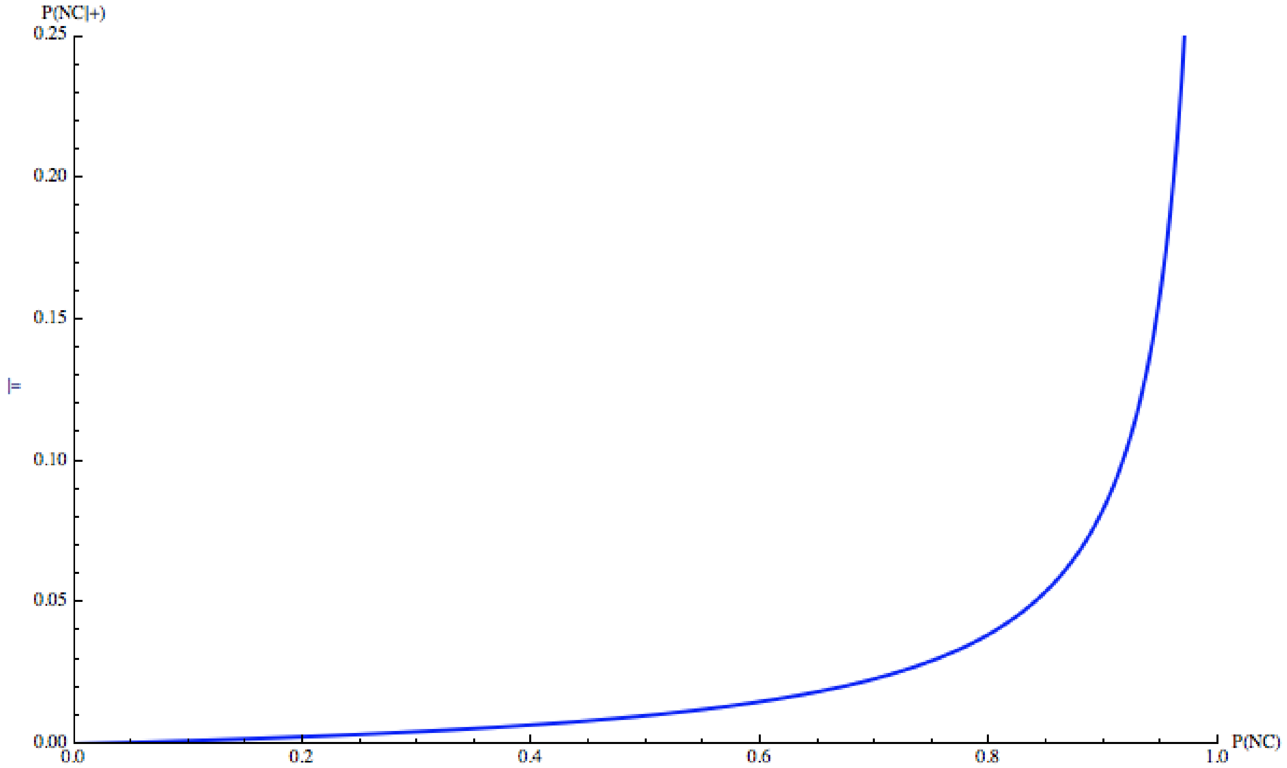

Figure 7: The Probability of a ”False Positive” Test for cancer as a function of Disease Incidence. Notice that for population groups that are extremely low risk – high $P(NC)$ – the probability of a ”false positive” is quite high!

$$P(NC|+) = \frac{(1-A)P(NC)}{(1-2A)P(NC)) + A} \tag{449}$$

Plotting this probability of a false positive as a function of $P(NC)$, we find a sharp increase as $P(NC)$ rises; which means that, for low risk groups – high $P(NC)$ – the probability of a false positive is quite high. This is an extremely important result for stressful diagnoses!

9 Parameter Estimation

9.1 Bayes Theorem, once again...

Now that we have introduced conditional probabilities, and “partitioning” our sample space into disjoint sets, we can now discuss the estimation of parameters. The estimation of parameters essentially boils down to a conditional probability, given some hypothesis. Let’s take our conditional probability structure from before:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{450}$$

$$= \frac{P(B|A)P(A)}{P(B)}, \tag{451}$$

or, Bayes’ basic theorem. Let us now call the event A the outcome that some parameter is correct – for example the gravitational acceleration constant. Let us call the event B the observation of data – say, the time it takes a rubber ball to drop from a roof. A can be framed as a disjoint family of sets, or a proper partitioning of the sample space into different 'parameter hypotheses', H_i – these are all separate values of g, ranging from 0 to ∞. Let us rewrite B as D, for data:

$$P(H_i|D) \;=\; \frac{P(D|H_i)P(H_i)}{P(D)} \tag{452}$$

The first term on the right hand side, in the numerator ($P(D|H_i)$) is called a likelihood function. The second term $P(H_i)$ is called the prior, or, our former intuition of the probability of a certain parameter in parameter space. And from our discussion before, we know that we can describe the probability of seeing the data D, as the sum of conditional probabilities:

$$P(D) \;=\; \Sigma_i P(D|H_i)P(H_i) \tag{453}$$

So, we can now write

$$P(H_i|D) \;=\; \frac{P(D|H_i)P(H_i)}{\Sigma_i P(D|H_i)P(H_i)}, \tag{454}$$

Which is essentially a "parameter average" over the data. Integrating "away" this hypothesis in the numerator is a process called marginalization. More on this later.

9.2 Chi-Squared Statistic and the Likelihood function

What the $P(D|H_i)$ term is, in the numerator is essentially the likelihood of seeing our data, given our hypothesis. Let's take a quick example and say that we are observing some data that looks like Figure 9.2. We can fit such data with any model we choose. In this example we will use a quadratic fit, or a model that looks like:

$$f(x) = mx + qx^2 + b \tag{455}$$

We can model each and every data point x_i, y_i as being generated from such a model with some error term, e_i which is determined by some probability distribution.

$$f(x_i) \;=\; y_i = mx_i + qx_i^2 + b + e_i \tag{456}$$

Let us isolate the error term,

$$e_i \;=\; y_i - mx_i + qx_i^2 + b \tag{457}$$

Now, suppose we assume the errors to be statistically independent – a reasonable assumption – for their range of values to spread from $-\infty \rightarrow \infty$ – also reasonable, as it precludes systematics. Then it might be a good guess for us to say our error is described by a normal distribution:

$$e_i \;\sim\; f = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{e_i^2}{2\sigma^2}} \tag{458}$$

where, the variance quoted above is the variance in our input, x_i. The probability of measuring a single error from our model – assuming some values m, b, q – is

$$
\begin{aligned}
P(e_1|m,b,q) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{e_1^2}{2\sigma^2}} && (459) \\
&= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y_1 - mx_1 + qx_1^2 + b\right)^2}{2\sigma^2}} && (460)
\end{aligned}
$$

The probability for measuring our entire set of data then, is the product of all of these conditional probabilities, or Likelihood's:

$$
\begin{aligned}
P(D|m,q,b) &= \Pi_i^N P(e_i|m,b,q) && (461) \\
&= \frac{1}{(2\pi\sigma^2)^{N/2}} e^{-\frac{\left(y_1 - mx_1 + qx_1^2 + b\right)^2}{2\sigma^2}} e^{-\frac{\left(y_2 - mx_2 + qx_2^2 + b\right)^2}{2\sigma^2}} \cdots && (462) \\
\cdots \; & e^{-\frac{\left(y_{N-1} - mx_{N-1} + qx_{N-1}^2 + b\right)^2}{2\sigma^2}} e^{-\frac{\left(y_N - mx_N + qx_N^2 + b\right)^2}{2\sigma^2}} && (463) \\
& && (464)
\end{aligned}
$$

Taking log of this quantity to introduce some sanity, we find

$$
\begin{aligned}
\log\left(P(D|m,q,b)\right) &= \frac{N}{2}\log(2\pi\sigma^2)\sum_i^N \left[\frac{y_i - mx_i + qx_i^2 + b}{2\sigma^2}\right] && (465) \\
&= \frac{N}{2}\log(2\pi\sigma^2)\sum_i^N \left[\frac{y_i - f(x_i)}{2\sigma^2}\right] && (466)
\end{aligned}
$$

I HAVE SCREWED THIS UP!!! COME back to this...

What we have just constructed is something called the likelihood function, or, the probability of observing our data given some generative model.

The generative model had two parts: $f(x_i)$ and e_i. We must be able to describe the probability distribution of the error – or at least guess it. Other wise we cannot reconcile the notion of Likelihood of a list of error values. $\mathcal{L}(D|m,q,b)$ often replaces $\log(P(D|m,q,b))$ above, and we have also stumbled upon the common definition of the χ^2 statistic, which is:

$$
\chi^2 = \sum_i^N \left[\frac{y_i - f(x_i)}{2\sigma^2}\right]^2 \qquad (467)
$$

And so we've got

$$
\log\left(P(D|m,q,b)\right) = \frac{N}{2}\log(2\pi\sigma^2)\chi^2 \qquad (468)
$$

If we want to maximize our likelihood, we had better minimize our χ^2 value, which is often seen as a metric of "fit" for a model to data.

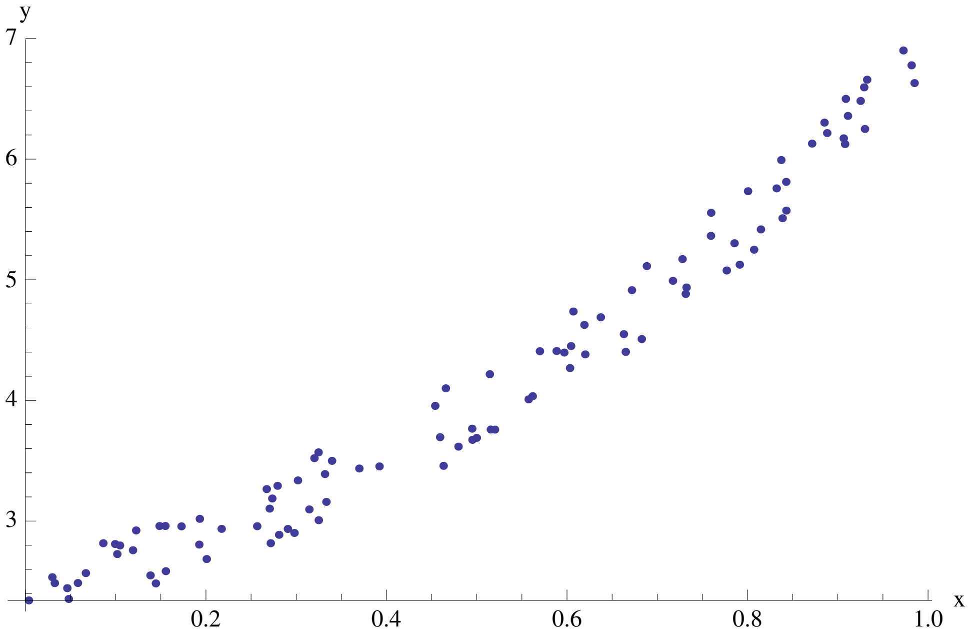

Figure 8: Some random data. Due to the curvature within the spread, one might like to do a least-χ^2 fit for a **quadratic** model– $f(x) = mx + qx^2 + b$

9.2.1 Estimating the Mean

The simplest example of an "estimation" of parameters, only involves estimating the mean of a sequence of numbers. Let's say we observe $X_1, X_2, X_3, \ldots, X_n$. Each of which are identical but statistically independent random variables. We "assume" a Gaussian distribution governs these random variables, with mean μ and variance σ^2 – or, what this is doing in terms of fitting to a model, we state that $f(x) = \mu$ and $f(x_i) = \mu + e_i$, where e_i is determined by a normal density – our likelihood function becomes:

$$\begin{aligned} P(D|\mu, \sigma^2) &= P(X_1, X_2, \ldots, X_n | X \sim N(\mu, \sigma^2)) \\ &\sim e^{\frac{(X_1-\mu)^2}{2\sigma^2}} e^{\frac{(X_2-\mu)^2}{2\sigma^2}} e^{\frac{(X_3-\mu)^2}{2\sigma^2}} \cdots e^{\frac{(X_n-\mu)^2}{2\sigma^2}} \\ &\sim e^{\sum \frac{(X_i-\mu)^2}{2\sigma^2}} \end{aligned}$$

We want to maximize the likelihood function for a given "guess" μ, so let us take the derivative with respect to our assumed μ and set it to zero. The value of μ that satisfies this, we will call $\hat{\mu}$, or our 'mean estimator'.

$$\begin{aligned} \log P(D|\mu, \sigma^2) &\sim -\sum \frac{(X_i - \mu)^2}{2\sigma^2} \\ \frac{\partial}{\partial \mu} \log P(D|\mu, \sigma^2) &\sim \sum \frac{(X_i - \mu)}{\sigma^2} \\ 0 &= \sum (X_i - \hat{\mu}) \\ N\hat{\mu} &= \sum (X_i) \\ \hat{\mu} &= \sum (X_i)/N \end{aligned}$$

This is our standard notion of average!!!

We can do the same thing for the variance, except this time we need to take into the factor outside our sum:

$$\begin{aligned}
\log\left(P(D|\mu,\sigma)\right) &= -\frac{\chi^2}{(2\pi\sigma^2)^{N/2}} \\
\frac{\partial}{\partial\sigma}\log\left(P(D|\mu,\sigma)\right) &= -\frac{\partial}{\partial\sigma}\left[\frac{1}{(2\pi\sigma^2)^{N/2}}\frac{\sum_i^N (X_i-\hat{\mu})^2 + n(\hat{\mu}-\mu)^2}{2\sigma^2}\right] = 0 \\
&= -\frac{\partial}{\partial\sigma}\left[\frac{1}{(2\pi\sigma^2)^{N/2}}\frac{\sum_i^N (X_i-\hat{\mu})^2 + n(\hat{\mu}-\mu)^2}{2\sigma^2}\right]
\end{aligned}$$

9.3 Least-χ^2 with matrices

Looking once again at Figure 9.2, we can

9.4 Correlation between two random variables

Let us say that we have two random variables X and Y described by the density functions f and g that are not statistically independent. There is some correlation. The correlation between these variables is closely related to the convolution of their separate density functions:

$$\begin{aligned}
\text{Corr}(x,y) &= \frac{\sum_{i=1}^N (x_i-\bar{x})(y_i-\bar{y})}{\sigma_x\sigma_y} &(469)\\
\text{Corr}(x,y) &= \frac{1}{\sigma_x\sigma_y}\int (x-m_{1x})(y-m_{1y})dxdy &(470)\\
\text{Corr}(x,y) &= \frac{1}{\sigma_x\sigma_y}\int (xy - m_{1y}x - m_{1x}y + m_{1x}m_{1y})dxdy &(471)\\
\text{Corr}(x,y) &= \frac{\langle xy\rangle - \langle x\rangle\langle y\rangle}{\sigma_x\sigma_y} &(472)
\end{aligned}$$

Notice that if the expectation value of the product is equal to the product of the expectation values, the correlation is zero and we therefore define the two random variables as being statistically independent. (This is a very similar statement to our non-intersecting sample spaces from before).

$$\begin{aligned}
P(x|y) &= \frac{P(x\cap y)}{P(x)} &(473)\\
\text{Corr}(x,y) &= P(x\cap y) - P(x)P(y) &(474)\\
&= (P(x|y) - P(y))P(x) &(475)
\end{aligned}$$

9.5 Orthogonality and Mutual Independence

For two statistically independent variables x and y, we say that there is no "overlap" between the sample spaces X and Y, or that the intersection of the sets is zero:

$$P(X\cap Y) = 0 \tag{476}$$

This means that the joint probability, which is the probability associated with the union of the two sample spaces X and Y, $P(X\cup Y)$, is the product of the probabilities:

$$P(X \cup Y) = P(X)P(Y) \tag{477}$$

This is a common property of statistically independent measurements – and the construction of Gaussian distributions. We can expand this out using set operations,

$$P(X \cup Y) = P(X)P(Y) + P(X \cap Y) \tag{478}$$

Or that, for statistically correlated observables x and y, we would expect to see joint probability $P(X \cap Y)$ higher than the product of the separate probabilities. But what does this mean in terms of our integral expressions earlier?

9.6 Estimator of Covariance

Working with results from last time, on the central limit theorem, we see that we can describe a Probability density function by it's fourier transform, the characteristic function

$$P(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}} \int d^d k e^{i\mathbf{k}\cdot\mathbf{x} - i\mathbf{k}\cdot\mathbf{c_1} - \mathbf{k}\cdot\mathbf{c_2}\cdot\mathbf{k} + \ldots} \tag{479}$$

Where, for now, I will ignore higher order terms corresponding to non-gaussianity. If we zero about the mean – absorbing our first cumulant, $\mathbf{c_1}$ into $\mathbf{x}$ – we find that we have a simple fourier transform of a Gaussian, which is itself a Gaussian.

$$P(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}} \int d^d k e^{i\mathbf{k}\cdot\mathbf{x} - \mathbf{k}\cdot\mathbf{c_2}\cdot\mathbf{k}} \tag{480}$$

$$P(\mathbf{x}) = \frac{1}{(2\pi)} \frac{1}{|\mathbf{c_2}|} e^{\mathbf{x}\cdot\mathbf{c_2^{-1}}\cdot\mathbf{x}} \tag{481}$$

Where $\mathbf{c_2}$ is the covariance matrix – rank (0,2) tensor – in k space.

If we are working with two random variables, x_1, x_2, we see that our k integration takes place over two dimensions, so we are left with

$$P(x_1, x_2) = \frac{1}{2\pi} \int dk_1 dk_2 e^{i(k_1 x_1 + k_2 x_2)} e^{-\frac{1}{2}\left(c_{11}k_1^2 + 2c_{12}k_1 k_2 + c_{22}k_2^2\right)} \tag{482}$$

Notice that our characteristic function has a simple form, which we can now play with in order to construct estimators for the correlation coefficient ρ commonly defined as

$$\rho = \frac{c_{12}}{\sigma_1 \sigma_2} \tag{483}$$

Let's take a look. First note that $c_{11} = \sigma_1^2$ and $c_{22} = \sigma_2^2$. These are the second cumulants of x_1 and x_2 respectively, their variance with respect to self. Re-writing our bivariate normal with these definitions, and assuming zero mean, (or $c_1 = c_2 = 0$), we find

$$\mathbf{c_2^{-1}} = \begin{pmatrix} \frac{1}{\sigma_1^2} & \frac{\rho}{\sigma_1\sigma_2} \\ \frac{\rho}{\sigma_1\sigma_2} & \frac{1}{\sigma_2^2} \end{pmatrix} \tag{484}$$

$$|\mathbf{c_2}| = \sigma_1^2 \sigma_2^2 (1 - \rho^2) \tag{485}$$

$$P(x_1,x_2) \;=\; \frac{1}{2\pi}\frac{1}{\sigma_1\sigma_2\sqrt{1-\rho^2}}^{-\frac{1}{2}\left(\frac{x_1^2}{\sigma_1^2}+2\rho\frac{x_1x_2}{\sigma_1\sigma_2}+\frac{x_2^2}{\sigma_2^2}\right)} \tag{486}$$

This last, nasty equation is the typical bivariate normal seen in the literahchurr. (literature in snoot-spell). But how the heck to get an estimator for the covariance?!?! We know that

$$\sigma_1^2 \;=\; \langle x_1^2\rangle - \langle x_1\rangle^2 = c_{11} \tag{487}$$

$$\;=\; -\frac{\partial^2 P}{\partial k_1^2}|_{(k_1,k_2)=(0,0)} + \left(\frac{\partial P}{\partial k_1}|_{(k_1,k_2)=(0,0)}\right)^2 \tag{488}$$

and

$$\sigma_2^2 \;=\; \langle x_2^2\rangle - \langle x_2\rangle^2 = c_{22} \tag{489}$$

$$\;=\; -\frac{\partial^2 P}{\partial k_2^2}|_{(k_1,k_2)=(0,0)} + \left(\frac{\partial P}{\partial k_2}|_{(k_1,k_2)=(0,0)}\right)^2 \tag{490}$$

The second term in both expansions – the first derivative of P(k) squared – is unnecessary, since we are assuming zero mean.

Let us show this trick of differentiating the characteristic function in order to get cumulant estimators explicitly, and then apply it to the covariance ρ

$$P(k_1,k_2) \;=\; \frac{1}{2\pi}\int dx_1dx_2e^{-i(k_1x_1+k_2x_2}P(x_1,x_2) = e^{-\frac{1}{2}\left(c_{11}k_1^2+2c_{12}k_1k_2+c_{22}k_2^2\right)} \tag{491}$$

Differentiating with respect to k_1 twice

$$\frac{\partial P}{\partial k_1} \;=\; \frac{1}{2\pi}\int dx_1dx_2 - ix_1e^{-i(k_1x_1+k_2x_2}P(x_1,x_2) = \left(c_{11}k_1 + c_{12}k_2\right)e^{-\frac{1}{2}\left(c_{11}k_1^2+2c_{12}k_1k_2+c_{22}k_2^2\right)} \tag{492}$$

$$\frac{\partial^2 P}{\partial k_1^2} \;=\; \frac{1}{2\pi}\int dx_1dx_2 - x_1^2e^{-i(k_1x_1+k_2x_2}P(x_1,x_2) = \left(c_{11}\right)e^{-\frac{1}{2}\left(c_{11}k_1^2+2c_{12}k_1k_2+c_{22}k_2^2\right)} \tag{493}$$

and setting $k_1 = k_2 = 0$, we find

$$\frac{\partial^2 P}{\partial k_1^2}|_{(k_1,k_2)=(0,0)} \;=\; \frac{1}{2\pi}\int dx_1dx_2 - x_1^2P(x_1,x_2) = \left(c_{11}\right) \tag{494}$$

$$c_{11} \;=\; \langle x_1^2\rangle \tag{495}$$

Now for the covariance, we can take the derivative with respect to k_1, then k_2, to get,

$$
\begin{aligned}
\frac{\partial P}{\partial k_1} &= \frac{1}{2\pi}\int dx_1 dx_2 - ix_1 e^{-i(k_1x_1+k_2x_2}P(x_1,x_2) = \left(c_{11}k_1 + c_{12}k_2\right)e^{-\frac{1}{2}\left(c_{11}k_1^2+2c_{12}k_1k_2+c_{22}k_2^2\right)} \\
\frac{\partial^2 P}{\partial k_1 \partial k_2} &= \frac{1}{2\pi}\int dx_1 dx_2 - x_1x_2 e^{-i(k_1x_1+k_2x_2}P(x_1,x_2) = \left(c_{12}\right)e^{-\frac{1}{2}\left(c_{11}k_1^2+2c_{12}k_1k_2+c_{22}k_2^2\right)} \\
\frac{\partial^2 P}{\partial k_1 \partial k_2}|_{(k_1,k_2)=(0,0)} &= \frac{1}{2\pi}\int dx_1 dx_2 - x_1x_2 P(x_1,x_2) = c_{12} \\
c_{12} &= \langle x_1 x_2 \rangle \\
\rho &= \frac{c_{12}}{\sigma_1\sigma_2} \\
\rho &= \frac{\langle x_1 x_2 \rangle}{\left(\langle x_1^2\rangle\langle x_2^2\rangle\right)^{1/2}}
\end{aligned}
$$

And voila! We have found the estimator – without taking into account bias – of the correlation coefficient!

Part III
Machine Learning and Handy Tricks

10 Cumulants of the Sample Mean

Let a random variable X be drawn from some probability distribution, $X \sim P(x)$. Then, the sum of N i.i.d. samples, or realizations is drawn from:

$$
\begin{aligned}
P(s_N = \sum_{n=1}^{N} x_n) &= (P * P * P \cdots P)(s_N) && (496) \\
&= \int dk e^{iks_N}\phi(k)^N = \int dk e^{iks_n + N\log\phi(k)} && (497) \\
&= \int dk e^{iks_N + N\psi(k)} && (498)
\end{aligned}
$$

This means that the equivalent cumulants for the sample mean – which is simply $\frac{s_N}{N}$ – are:

$$
\begin{aligned}
\bar{x} &= \sum_{n}^{N} \frac{x_n}{N} && (499) \\
\langle \bar{x}^k \rangle_c &= \frac{1}{N^k}\langle \left(\sum_n x_n\right)^k \rangle_c && (500)
\end{aligned}
$$

Expanding out the term inside the brackets using the multinomial theorem, we get:

$$
\begin{aligned}
\langle \bar{x}^k \rangle_c &= \frac{1}{N^k}\langle \sum_{\vec{k}} \frac{k!}{k_1!k_2! \cdot k_N!}\prod_{n=1}^{N} x_n^{k_n}\rangle_c && (501) \\
&= \frac{1}{N^k}\sum_{\vec{k}} \left(\begin{array}{c} k \\ k_1 \cdots k_N \end{array} \right) \langle \prod_{n=1}^{N} x_n^{k_n} \rangle_c && (502)
\end{aligned}
$$

any dependence between the samples x_n yields this expression nontrivial, but if we assume i.i.d. we just get:

$$\langle \bar{x}^k \rangle_c = \frac{1}{N^{k-1}} \langle x^k \rangle_c \tag{503}$$

Where the term in the RHS brackets is just a single realization of our random variable x. Why is this important? Because it tells us that the variance, which is the $k = 2$ cumulant, scales as $\frac{1}{N}$, and the skewness, which is defined as:

$$\text{skew} = \frac{\langle \bar{x}^3 \rangle_c}{(\langle \bar{x}^2 \rangle_c)^{3/2}} \sim \frac{1}{\sqrt{N}} \tag{504}$$

scales as one over square root N. This means that our distribution on the mean collapses to a Gaussian **at least** as fast as $\frac{1}{\sqrt{N}}$, given some initial asymmetry in the distribution of x. (If we have symmetry, things are even better and we need only worry about the kurtosis, which goes like $\frac{1}{N}$.)

Such considerations are important when you ask yourself: at what point can I consider the estimator of the mean to be drawn from a Gaussian? How does my approximation scale with sample size? These are very important questions in the real world, and it nice to have a sense of what's holding you back from being exact – namely, that $\frac{1}{\sqrt{N}}$ for skewness, which goes along with a hermite polynomial, and $\frac{1}{N}$, which goes with another.)

11 Chernoff Bound

Let's say we are trying to estimate the base rates of some classes $C = \{1, 2, 3 \dots K\}$. After polling N people, each of which will elect one of theses classes, how confident are we in our results, $P(c)$? The first step to answering this question is to simplify the problem and treat it as a sequence of binomials.

Let us at first restrict ourself to two classes. $c = 1$ means "no" or "Democrat" and $c = 0$ means "yes", or "Republican". If weuse a bernouli random variable $X = 0, 1$ to represent a single "vote" we get

$$P(X|p) = p^{I_{x=1}} (1-p)^{I_{x \neq 1}} \tag{505}$$

$$q = 1 - p \tag{506}$$

Where I is the indicator function. The expectation of this distribution is:

$$\langle X \rangle = p \tag{507}$$

$$\langle X^2 \rangle_c = \text{Var}(X) = pq \tag{508}$$

Given these first two cumulants, we can write the Markov bound:

$$P(X > a) \leq \frac{\langle X \rangle}{a} \tag{509}$$

$$\leq \tfrac{p}{a} \tag{510}$$

and the improved Chebyshev bound:

$$P(|X - p| \geq mpq) \leq \frac{pq}{mpq} = \frac{1}{m} \tag{511}$$

Similarly, for a binomial random variable, a sequence of votes:

$$
\begin{aligned}
X_N &= \sum_{i=1}^{N} X_i && (512)\\
X_N &\sim \left(\begin{array}{c} N \\ X_N \end{array}\right) p^{X_N}(1-p)^{X_N} && (513)
\end{aligned}
$$

we get expectation and variance

$$
\begin{aligned}
\langle X_N \rangle &= Np && (514)\\
\langle X_N^2 \rangle_c = \mathrm{Var}(X) &= Npq && (515)
\end{aligned}
$$

So our bound becomes

$$
P(|X_N - Np| \geq mNpq) \leq \frac{1}{m} \tag{516}
$$

But that seems repetitive. Let's define a new statistic, the estimator of the mean:

$$
\begin{aligned}
\hat{\mu}_N &= \frac{X_N}{N} = \frac{\sum_{i=1}^{N}}{N} && (517)\\
\langle \hat{\mu}_N \rangle &= p && (518)\\
\langle \hat{\mu}_N^2 \rangle &= \frac{pq}{N} && (519)
\end{aligned}
$$

Now we have an inequality that scales with the number of votes, or the survey size, N:

$$
P(|\frac{X_N}{N} - p| \geq a) \leq \frac{pq}{Na^2} \tag{520}
$$

More generally the estimator of any mean, can be written

$$
P(|\frac{X_N}{N} - \mu| \geq \epsilon) = \frac{\sigma^2}{N\epsilon^2} \tag{521}
$$

So, we can now make a few helpful statements. Our confidence that the estimator of the mean is within plus or minus ϵ of the true value is precisely $1 - P(|\frac{X_N}{N} - \mu| \geq \epsilon)$, for a "survey" of N participants. So let's see how accuracy ϵ scales with survey size and confidence. Writing confidence as:

$$
\begin{aligned}
\text{confidence} = 1 - \delta &= 1 - P(|\frac{X_N}{N} - \mu| \geq \epsilon) && (522)\\
\delta &\leq \frac{\sigma^2}{N\epsilon^2} && (523)\\
\epsilon &\leq \frac{\sigma}{\sqrt{N\delta}} && (524)
\end{aligned}
$$

So we that accuracy scales like one over square root survey size, and that if we want to be twice as confident with a given accuracy, we'd better poll twice as many people. Sometimes in the real world, this implies a cost – at least in terms of time! – and so we'd like to do even better, if at all possible.

It turns out we can do much, much better by using something called the Chernoff Bound, which assumes not only pair-wise independence between our random votes X_i – as we assumed by writing $\mathrm{Var}(\frac{X_N}{N}) = \frac{\sigma^2}{N}$ – but independence across all the variables, for all possible combinations.

How do we do this? Re-write the markov bound with some free parameter:

$$P(x \geq a) \;=\; P(e^{sx} \geq e^{as}) \leq \frac{\langle e^{sx} \rangle}{e^{sa}} \tag{525}$$

We see that the numerator is simply the moment generating function, evaluated at s. If we want to swap the inequality, we just send $s \to -s$:

$$P(x \leq a) \;=\; P(e^{-sx} \geq e^{-as}) \leq \frac{\langle e^{-sx} \rangle}{e^{-sa}} \tag{526}$$

For our bernouli distribution on a single “vote”, we have:

$$\begin{aligned} M(s) = \langle e^{sX_1} \rangle &= \sum_X e^{sX} P(X) && (527)\\ &= e^s p + 1 - p && (528)\\ M(s) &= p(e^s - 1) + 1 && (529) \end{aligned}$$

Now, if we write down the moment generating function of the sum, we find, if the X_i’s are all independent:

$$\begin{aligned} \langle e^{sX_N} \rangle &= e^{s\sum_i X_i} && (530)\\ &= \langle \prod_{i=1}^{N} e^{sX_i} \rangle && (531)\\ M_N(s) &= M(s)^N && (532)\\ &= (1 + p(e^{s_N} - 1))^N && (533) \end{aligned}$$

Recall that we want to bound the probability of our random variables X_N being outside some range, so we want to minimize the right handside of this equation:

$$P(X_N \geq a) \;=\; M_N(s) e^{-sa} \tag{534}$$

Minimizing this equation is the same is minimizing the log, since both sides are positive definite:

$$\log P(X_N \geq a) \;=\; N \log\left(1 + p(e^{s_N} - 1)\right) - sa \tag{535}$$

Taking the derivative with respect to s and setting things equal to zero, we get:

$$s \;=\; \log\left(\frac{a}{Np}\right) \tag{536}$$

Which, for arbitary mean, is the same thing as saying

$$s \;=\; \log\left(\frac{a}{\mu}\right) \tag{537}$$

So now let's choose $a = \mu(1+\theta)$, so that θ is our fractional accuracy relative to the variable and $s = \log(1+\delta)$. Putting this into our equation gives:

$$\log P(X \geq \mu(1+\theta) \leq \mu\left(\theta - (1+\theta)\log(1+\theta)\right) \tag{538}$$

$$P(X \geq \mu(1+\theta) \leq \left(\frac{e^\theta}{(1+\theta)^{1+\theta}}\right)^\mu \tag{539}$$

But, we could have taylor expanded the logarithm above, to get:

$$\begin{aligned}\log P(X \geq \mu(1+\theta) &\leq \mu\left(\theta - (1+\theta)\log(1+\theta)\right) \leq \mu(-\theta^2/3) \\ P(X \geq \mu(1+\theta) &\leq e^{-\theta^2\mu/3}\end{aligned}$$

So this is the upper bound. What about the lower? We just send $s \to -s$ to get

$$P(X \leq \mu(1-\theta) \leq e^{-\theta^2\mu/2} \tag{540}$$

So the lower bound, with the same taylor expansion, does slightly better! We can now make a double sided bound:

$$P(|X-\mu| \geq \mu\theta) \leq 2e^{-\theta^2\mu/3} \tag{541}$$

If we specify, once again $1-\delta = 1 - P(|X-\mu| \geq \mu\theta)$ as our survey confidence, we can write

$$P(|X_N - pN| \geq Np\theta) \leq 2e^{-\theta^2 Np/3}$$

Let $\theta = \frac{\epsilon}{p}$

$$P(|X_N - pN| \geq N\epsilon) = P(|\frac{X_N}{N} - p| \geq \epsilon) \leq 2e^{-\frac{N\epsilon^2}{3p}} \tag{542}$$

Now for a few tricks,

$$\delta = P(|\frac{X_N}{N} - p| \geq \epsilon) \tag{543}$$

$$\delta \leq 2e^{-\frac{N\epsilon^2}{3p}} \leq 2e^{-\frac{N\epsilon^2}{3}} \tag{544}$$

$$e^{N\epsilon^2/3} \geq \frac{2}{\delta} \tag{545}$$

$$N \geq \frac{3}{\epsilon^2}\log(\frac{2}{\delta}) \tag{546}$$

This result is known as the sampling theorem. In order to constrain our underlying "base rate" to $p = X_N/N \pm \epsilon$ with confidence $1-\delta$, we need to poll AT LEAST N people. This is an extremely useful result, and has much better scaling properties than the Chebyshev bound, which would suggest:

$$N \geq \frac{1}{\delta}\frac{\sigma^2}{\epsilon^2} \tag{547}$$

We see that we pay just as heavily for accuracy ϵ but much more heavily for confidence δ. We would like to make δ as small as possible, and so if we taylor expand both expressions, and call $R = 1 - \delta$ confidence, we find:

$$\begin{aligned} \text{Chebyshev}: N &\sim 3R\frac{1}{\epsilon^2} && (548) \\ \text{Chernoff}: N &\sim 3\log(2R)\frac{1}{\epsilon^2} && (549) \end{aligned}$$

The latter CH is superior – logarithmic scaling in confidence rather than linear!

Such considerations are important, because often we are interested in constraining the maximum error of many different random variables – say, K class base rates, $P(c) = p_c$ – with the same accuracy ϵ. This can be accomplished by the union bound:

$$P(X_1 \cup X_2 \cup X_3 \cdots X_N) \leq \sum_{i=1}^{N} P(X_i) \tag{550}$$

So that, if we want to write the probability that any of our mean base rate estimates, for some "poll" with K possibilities is incorrect by a factor greater than ϵ, we can write:

$$\begin{aligned} P(|\hat{\mu}_{N_1} - p_1| \geq \epsilon \cup |\hat{\mu}_{N_2} - p_2| \geq \epsilon \cup \ldots |\hat{\mu}_{N_K} - p_K| \geq \epsilon) &\leq \sum_{c=1}^{K} P(|\hat{\mu}_{N_c} - p_c| \geq \epsilon) \\ \delta &\leq \sum_{c=1}^{K} \delta_c \end{aligned}$$

Assuming the same confidence for every independent class bound, we get:

$$\delta \leq \sum_{c=1}^{K} \delta_c \leq 2Ke^{-\theta^2\mu/3} \tag{551}$$

which implies:

$$N \geq \frac{3}{\epsilon^2}\log(\frac{2K}{\delta}) \tag{552}$$

So to bound the maximum of K different statistics, or base rates with the same accuracy ϵ, with some confidence δ, we essentially send $\delta \to K\delta$ from a single binomial test; and, since the Chernoff scales logarithmically, this is not such a big deal!

12 Logistic Regression and Naive Bayes Classifier

Let's say that we see some features in a dataset, $X = \{\vec{x}\}$ along with categorical variables, $\{c\}$, which we'll call a "class". This class could be male/female, the features could be, say, height, weight, favorite ice cream, etc. Normally in classification problems, we are interested in predicting c given $\vec{x}$. Or more accurately the conditional probability $P(c|\vec{x})$. How do we go about this?

Well, there are two ways to approach the problem. One is called discriminative, the other generativ – they just correspond to breaking down the joint distribution of class and features, $P(c, \vec{x})$ differently. Let's look at the following picture, which in machine learning is called a directed acyclic graph, or DAG:

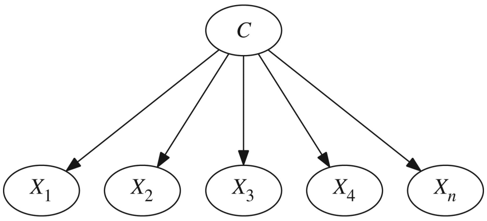

Figure 9: A generative model

The arrows represent "cause", in the sense that $x_1, x_2, \dots x_n$ all are independent, given c. A DAG in machine learning informs the structure of the joint probability distribution on $P(c, \vec{x})$. We can write from that picture, by chain rule:

$$P(c, \vec{x}) \quad = \quad P(c)P(x_1|c)P(x_2|c) \cdots P(x_n|c) \tag{553}$$

So we see that the class-conditional distribution is given by:

$$P(\vec{x}|c) \quad = \quad)P(x_1|c)P(x_2|c) \cdots P(x_n|c) \tag{554}$$

The "product of the marginals". This type of framework is called common cause. We have a common cause to all of our features, which is tantamount to saying that the variables $x_1 \dots x_n$ have no interdependence once we know the class. Such a framework can be used to get the object we are interested in, $P(c|\vec{x})$, by using Bayes rule:

$$P(c|\vec{x}) \quad = \quad \frac{P(c)P(\vec{x}|c)}{P(\vec{x})} \tag{555}$$

$$\quad = \quad \frac{P(c)\prod_{i=1}^{n} P(x_i|c)}{\sum_{c'} P(c')\prod_{i=1}^{n} P(x_i|c')} \tag{556}$$

When trying to figure out which class our data point belongs in, given some features, we simply choose the maximum of the numerator:

$$\text{guess} \quad = \quad \text{argmax}_c \left(P(c) \prod_{i=1}^{n} P(x_i|c) \right) \tag{557}$$

This is called the Naive Bayes classifier, and stems exactly from the DAG we wrote down before. Now, that's all well and good, Naive Bayes has some nice computational properties in that it can be learned/updated continuously, but what if we reversed the arrows on our DAG, above? Then we would have what's called a common effect framework, for which, even if we have the same structure of our joint distribution, we find that conditioning on class INDUCES correlation between the features x_i. One can see this by just looking at two features. We have:

$$P(x_1, x_2, c) \quad = \quad P(c|x_1, x_2)P(x_1)P(x_2) \tag{558}$$

conditioning on class we get a probability that cannot be factorized or separated in x_1, x_2:

$$P(x_1, x_2|c) \;=\; \frac{P(c|x_1, x_2)P(x_1)P(x_2)}{\int dx_1 dx_2 P(c|x_1, x_2)P(x_1)P(x_2)} \tag{559}$$

This expression is typical of a "common effect" model, where conditioning on class induces effects between the likelihood of different features. I won't get too much into this now, but in the common cause graph we could write:

$$x_i \perp x_j|c \qquad \forall\, i \neq j \tag{560}$$

Or "class conditional independence" between features. For the common effect, we cannot write such a thing. The common effect framework is often called "explaining away", but I won't get into that either. The point is, we can relate the Naive Bayes classifier, a generative framework, directly to the discriminative or logistic classificier, which has the same DAG structure but with the arrows reversed. We write:

$$P(\vec{x}|c) \;\sim\; \prod_{i=1}^{n} P(x_i|c) \tag{561}$$

and, for probability distributions in the exponential family we can simply write this as:

$$P(\vec{x}|c, \theta) \;=\; \exp\left(\theta_c \cdot \mathbf{x} - Z(c)\right) h(\mathbf{x}) \tag{562}$$

Where $Z(c)$ is the class partition function, written as:

$$Z(c) \;=\; \log\left(\int dx e^{\theta_c \cdot \mathbf{x}} h(\mathbf{x})\right) \tag{563}$$

A physicist might look at this and notice that the log sum integral function is eerily reminiscent of the Free energy $\mathcal{F}$. But more on that later. It turns out that the maximum likelihood parameters of θ that I have written above, are precisely those that maximize entropy – thereby minimizing free energy – while also satisfying data constraints on "sufficient statistics" for $\mathbf{x}$, such that the empirical average is equal to the theoretical average. One can show this by maximizing the likelihood:

$$\mathcal{L}(\{c_n, \mathbf{x}_n\}_{n=1}^N|\theta) \;=\; \exp\left(\sum_n \theta_c \cdot \mathbf{x}_n - NZ(c)\right) \tag{564}$$

$$\log \mathcal{L}(\{c_n, \mathbf{x}_n\}_{n=1}^N|\theta) \;=\; \sum_n \theta_c \cdot \mathbf{x}_n - NZ(c|\theta) \tag{565}$$

Taking the derivative with respect to θ we see that we get:

$$\frac{\partial \mathcal{L}}{\partial \theta} \;=\; \sum_n \frac{\mathbf{x}_n}{N} - \frac{\partial Z(c|\theta)}{\partial \theta} \tag{566}$$

But the second term on the right is just the moment of the feature $\mathbf{x}$ under our distribution:

$$\frac{\partial Z(c|\theta)}{\partial \theta} \;=\; \int dx \left(e^{\theta_c \cdot \mathbf{x} - Z(c)} \mathbf{x}\right) = \langle \mathbf{x} \rangle_\theta \tag{567}$$

So we find that the maximum likelihood estimate θ is such that the empirical mean of the data equals the mean under our distribution.

$$\frac{\partial \mathcal{L}}{\partial \theta} = 0 \;\Rightarrow\; \sum_n \frac{\mathbf{x}_n}{N} = \langle \mathbf{x} \rangle_\theta \tag{568}$$

Such an exponential family representation is very convenient for DAG's, because we can write down separable joint distributions as a sum of neighbor-wise interaction energies in our 'Hamiltionian" :

$$P \;\sim\; \exp\left(\mathcal{H}(\vec{x})\right) \sim \exp\left(\sum \mathcal{H}_n(x_{n+1}, x_{n-1}, x_n)\right) \tag{569}$$

If we treat our sufficient statistics as simply the conditional mean:

$$T_c(\mathbf{x}) \;=\; \mathbf{x}1_{c=c'} \tag{570}$$

where I've used the $1_{c=c'}$ indicator function above to restrict the mean to each class separately, we get linear discriminant logistic regression, or the softmax function:

$$\begin{aligned} P(c|\vec{x}) &= P(c)P(\vec{x}|c)/P(\vec{x}) && (571)\\ &= \frac{\exp\left(\theta \cdot T_c(\mathbf{x}) - Z(c)\right) h(\mathbf{x}) P(c)}{\sum_{c'} \exp\left(\theta \cdot T_c(\mathbf{x}) - Z(c')\right) h(\mathbf{x}) P(c')} && (572)\\ &= \frac{\exp\left(\theta \cdot T_c(\mathbf{x})\right)}{\sum_{c'} \exp\left(\theta \cdot T_{c'}(\mathbf{x})\right)} && (573) \end{aligned}$$

As we change the form of our sufficient statistics $T(\vec{x})$, one can see that we change the shape of our decision boundary. This framework is the discriminative one, which has distinct advantages in that it implicitly takes into account interdependence/correlation of the feature $\vec{x}$. Andrew Ng calls Naive Bayes and Logistic Regression a discriminative-generative pair.

13 Learning Theory

After a somewhat embarrassing moment on Kaggle.com, where I blatantly overfit a training set X with a boosted decision tree – XGboost – and then got a horrible, horrible, generalized result, I thought it'd be time to consider model complexity and learning theory, and in general the debate that has been bubbling beneath the Machine Learning and Statistics community.

Recently, there has been a lot of success using highly flexible models, such as boosted regressors, classifiers, decisions trees, random forests, etc. to model very complex data sets. (One should lump neural networks in here, too, since they perform highly non-trivial, non-linear transformation to inputs.) Leo Breiman wrote a very interesting paper in 2001 called "Statistical Modeling: The two Cultures" which highlights – and insults – the attachment of old school statisticians to generative models, where the data observed is characterized/explained as being generated by some "process". For instance, we say that the obesity in the United States is generated by a linear superposition of many factors, say income, age, sex, etc.:

$$\text{obsesity} = f(\mathbf{x}) \;=\; \sum_n \theta_n x_n \tag{574}$$

Which is of course a very stupid assumption, since we know there are non-linear interactions between these features of a person, that create complex results. More flexible models can capture this, by multiplying the different features in some polynomial. Say for instance, we add all quadratic – second order – terms:

$$f(x) = \theta \cdot \mathbf{x} + \mathbf{x} \cdot A \cdot \mathbf{x} \quad (575)$$

Where A is some feature mixing matrix. There are way more parameters in our model now, and some will prove to be important, while others will not. These days, the **most** old school statisticians – it seems to me – expand features as polynomials, like above, and then use methods to cut down on the model complexity, getting rid of combinations and or linear features that are not deterministic of our outcome. This is all well and good. One can use the precise framework of $p-$values and hypotheses tests, etc. to characterize fit, but Leo Breiman says that by using such methods – and I don't pretend to have exhausted all such models he pointed a finger at, or properly explained them – we really limit ourselves. Essentially, the gist is that if we get away from this generative framework and look more towards flexible algorithms, such as Random forests and boosted decision trees, we can have greater predictive power in the real world.

If Kaggle competitions are treated as valid evidence, this is certainly true, since virtually every winner of these competitions – outside computer vision – has used gradient boosted decision trees or Random forests explicitly for their solution or as part of a larger ensemble of models. The trouble is, how do we tune these models to not overfit? How do we interpret and regularize them? Constraining a linear or logistic regression to have small weights with a quadratic penalty term is essentially the same thing **imposing** entropy, or some simplicity measure on the model (which turns out to be a Gaussian prior); but for trees and forests, how does one characterize the degrees of freedom? This is the trouble I had over the weekend, trying to fit a very complex algorithm that I didn't understand to a dataset and then getting burned for it. The nice thing about Logistic Regression is that those linear weights θ can be read as importance values – when taking the absolute value: they tell you how much each feature – which, hopefully has been properly scaled – contributes to a decision, or pushing towards a decision in feature space $\mathbf{x}$. (I've talked about this before.) Trees are much harder, and although the choice of split in classification problems, based on the gini index or information gain are useful, they are only a piece of the story. The first split of a decision tree can be viewed as the most important feature of course, but one clever comment I read in an AI book is that pruning trees is much better than limiting their depth, as what often happens, for instance with and XOR classification problem, is you split on x_1, getting not so great results, but then the following split on x_2 is highly deterministic, implying that the operator x_1 AND x_2 is a very important feature, which would have been missed had the tree not been able to "grow". Clever as this comment is when it comes to training and making practical decisions when using Decision trees, it also highlights the fact that feature importance in a Decision tree can be LATENT, or, difficult to comprehend. Sometimes the logical conjunction of features or splits is what's important, and that will not show up so easily to the eye when looking at a decision tree.

So, just for fun and practice with the Hoeffding, Chernoff inequalities, and the Union bound, let's take a look at some elementary learning theory (I'm lifting this directly from Andrew Ng's notes at Stanford). Say we are in binary classification problem, where are data is a sequence of features $\mathbf{x}$ and corresponding classes y:

$$D = \; = \{\mathbf{x}_n, y_n\}_{n=1}^{N} \quad (576)$$

We would like to estimate our generalization error – which is just a specific type of loss function that we're going to treat as a conditional random variable:

$$\epsilon(h(x)) = 1_{h(x)\neq y} \quad (577)$$

Where 1 is the indication function. The expectation value of the generalization error is characterized by a probability distribution, which we know nothing about:

$$\epsilon(h(x)) \sim P(\epsilon(h(x))|D) \tag{578}$$

But, we can estimate the mean of this distribution, given some samples in D by using the Hoeffding bound. The greatest $\epsilon(h(x))$ can be is unity for a given x, and the lowest it can be is zero. This means we can write:

$$P(|\epsilon(h(x)) - \sum_{n=1}^{N} \frac{\epsilon(h(x_n))}{N}| \geq \gamma) \leq 2e^{-2N\gamma^2} \tag{579}$$

where the nasty term in the argument of our probability distribution is just the mean of our errors, what's called the Empirical risk:

$$\hat{\epsilon}(h(x)) = \sum_{n=1}^{N} \frac{\epsilon(h(x_n))}{N} \tag{580}$$

Writing this a bit more simply, then, this is really just the standard problem of estimating the mean of a binomial distribution, for example – but we haven't specificied the PDF, only that all of the samples x_n, y_n are i.i.d!

$$P(|\epsilon(h(x)) - \hat{\epsilon}(h(x))| \geq \gamma) \leq 2e^{-2N\gamma^2} \tag{581}$$

We can see that as $N \to \infty$ – an infinite training set – we are going to be spot on: our training error $\hat{\epsilon}$ will be precisely the same as our generalization error ϵ. But, recall that when fitting models, $h(x)$ is actually parametrized by some θ. So what we are really worried about is the probability that ANY of our models' train set error, $\hat{\epsilon}(h_\theta(x))$ deviate far from their generalization error. For such a thing we use the union bound:

$$P(A \cup B \cup C \dots) \leq P(A) + P(B) + P(C) + \dots \tag{582}$$

Now, to make life simpler, let's say our model space is discrete, and we really only have a choice of K θs. Then we've got:

$$P(\cup_{k=1}^{K} |\epsilon(h_k(x)) - \hat{\epsilon}(h_k(x))| \geq \gamma) \leq 2Ke^{2N\gamma^2} \tag{583}$$

So, as we've seen before. Things aren't that much worse. Let's write the probability that our training error is more than γ different from our generalization error as δ, and then we can state:

$$\gamma \leq \sqrt{\frac{\log(2K/\delta)}{2N}} \tag{584}$$

$$N \geq \frac{1}{2\gamma^2} \log\left(\frac{2K}{\delta}\right) \tag{585}$$

What do these two statements mean? The first that our estimation of generalization error gets tighter as N gets large. More specifically, it scales at worst like $\frac{1}{\sqrt{N}}$, just Monte Carlo Models. We also see that accuracy goes logarithmic with confidence, which is just a remnant of usual Hoeffding/Chernoff bound statements. In the second line, we see that the requisite number of Training examples, for some given confidence δ and accuracy on empirical risk γ, goes logarithmic with the degrees of freedom. A priori, this seams very nice, but let's develop things further.

When we're given a training set, we're not gauranteed to fit the optimal model, but we know our empirical risk estimate is at most 2γ away the OPTIMAL empirical risk. Let's show this. First, call our fitted model $h_{\hat{\theta}}$, and the optimal one h_{θ^*}. We have, from the above Hoeffding/Chernoff inequality, for a single fitted model:

$$\epsilon(h_{\hat{\theta}}) \quad \leq \quad \hat{\epsilon}(h_{\hat{\theta}}) + \gamma \tag{586}$$

This is just comparing generalization risk to training risk. Now let's compare to the optimal, by noting that, if we fit to the wrong model, $\hat{\epsilon}(h_{\hat{\theta}}) \leq \hat{\epsilon}(h_{\theta^*})$ – otherwise we would have chosen something else! Then we can write:

$$\epsilon(h_{\hat{\theta}}) \quad \leq \quad \hat{\epsilon}(h_{\theta^*}) + \gamma \tag{587}$$

and, now converting the training risk to the generalization risk on the optimal model, we get:

$$\epsilon(h_{\hat{\theta}}) \quad \leq \quad \epsilon(h_{\theta^*}) + 2\gamma \tag{588}$$

This is a very nice inequality, because what it's basically saying is that our empirical risk is less than the optimal fit, plus some accuracy term:

$$\epsilon(h_{\hat{\theta}}) \quad \leq \quad \left(\min_{h_\theta} \epsilon(h_\theta(X))\right) + \sqrt{\frac{2\log\left(\frac{2K}{\delta}\right)}{N}} \tag{589}$$

With highly complex models, the first term on the right hand side will be quite small – low bias – since we can fit our data almost exactly. But conversely, in such a case K becomes large and we have no good hold on how our model will extend into the real world – high variance. How bad does this scaling of complexity go? A rough argument from Andrew Ng's notes is that for floating point numbers, for D parameters, we have approximately:

$$K \quad = \quad 2^{64D} \tag{590}$$

and so our inequality goes like:

$$\epsilon(h_{\hat{\theta}}) \quad \leq \quad \left(\min_{h_\theta} \epsilon(h_\theta(X))\right) + \sqrt{\frac{2D\log\left(\frac{2^{65}}{\delta}\right)}{N}} \tag{591}$$

and so the requisite number of training examples – for a given accuracy γ and confidence δ – goes linearly with parameter dimension.

But this is not entirely correct. One can use something called the vapnik-chervonenkis dimension, which I won't get into here, maybe in a later post, to characterize the complexity of a model. This is all a very active part of research, and it obviously is quite difficult to get a handle on how well an algorithmic solution will generalize, once trained on a training set.

For instance, what is K for gradient boosted decision tree? How does one characterize its vapnik-chervonenkis dimension? This question is very important if you want to fit such models to data, rather than being a blockhead and just using Randomized Grid Search for Hyper-Parameters in python.

14 Bias Variance Tradeoff

After

15 DAG's

Let's say that we see some features in a dataset, $X = \{\vec{x}\}$ along with categorical variables, $\{c\}$, which we'll call a "class". This class could be male/female, the features could be, say, height, weight, favorite ice cream, etc. Normally in classification problems, we are interested in predicting c given $\vec{x}$. Or more accurately the conditional probability $P(c|\vec{x})$. How do we go about this?

Well, there are two ways to approach the problem. One is called discriminative, the other generativ – they just correspond to breaking down the joint distribution of class and features, $P(c, \vec{x})$ differently. Let's look at the following picture, which in machine learning is called a directed acyclic graph, or DAG:

The arrows represent "cause", in the sense that $x_1, x_2, \ldots x_n$ all are independent, given c. A DAG in machine learning informs the structure of the joint probability distribution on $P(c, \vec{x})$. We can write from that picture, by chain rule:

$$P(c, \vec{x}) = P(c)P(x_1|c)P(x_2|c) \cdots P(x_n|c) \tag{592}$$

So we see that the class-conditional distribution is given by:

$$P(\vec{x}|c) =)P(x_1|c)P(x_2|c) \cdots P(x_n|c) \tag{593}$$

The "product of the marginals". This type of framework is called common cause. We have a common cause to all of our features, which is tantamount to saying that the variables $x_1 \ldots x_n$ have no interdependence once we know the class. Such a framework can be used to get the object we are interested in, $P(c|\vec{x})$, by using Bayes rule:

$$\begin{aligned} P(c|\vec{x}) &= \frac{P(c)P(\vec{x}|c)}{P(\vec{x})} && (594) \\ &= \frac{P(c)\prod_{i=1}^n P(x_i|c)}{\sum_{c'} P(c')\prod_{i=1}^n P(x_i|c')} && (595) \end{aligned}$$

When trying to figure out which class our data point belongs in, given some features, we simply choose the maximum of the numerator:

$$\text{guess} = \text{argmax}_c \left(P(c) \prod_{i=1}^n P(x_i|c) \right) \tag{596}$$

This is called the Naive Bayes classifier, and stems exactly from the DAG we wrote down before. Now, that's all well and good, Naive Bayes has some nice computational properties in that it can be learned/updated continuously, but what if we reversed the arrows on our DAG, above? Then we would have what's called a common effect framework, for which, even if we have the same structure of our joint distribution, we find that conditioning on class INDUCES CORRELATION between the features x_i. One can see this by just looking at two features. We have:

$$P(x_1, x_2, c) = P(c|x_1, x_2)P(x_1)P(x_2) \tag{597}$$

conditioning on class we get a probability that cannot be factorized (separable in x_1, x_2):

$$P(x_1, x_2|c) = \frac{P(c|x_1, x_2)P(x_1)P(x_2)}{\int dx_1 dx_2 P(c|x_1, x_2)P(x_1)P(x_2)} \tag{598}$$

This expression is typical of a "common effect" model, where conditioning on class induces creates interactions between features. I won't get too much into this now, but in the common cause – generative – framework we could write:

$$x_i \perp x_j|c \qquad \forall\, i \neq j \tag{599}$$

Or have "class conditional independence" between features. For the common effect – or discriminative – framework we cannot write such a thing. What we can do, however, for any DAG, is frame things through an a PDF in the exponential family, which takes the form:

$$P(\vec{x}|c) = \exp\left(\theta \cdot T(\mathbf{x}) - Z(c)\right) h(\mathbf{x}) \tag{600}$$

Where $Z(c)$ is the class partition function, written as:

$$Z(c) = \log\left(\int dx e^{\theta \cdot T(\mathbf{x})} h(\mathbf{x})\right) \tag{601}$$

A physicist might look at this and notice that the log sum exp function is eerily reminiscent of the Free energy $\mathcal{F}$, the legendre transform of entropy. It turns out that the maximum likelihood parameters of θ that I have written above, are precisely those that maximize entropy – thereby minimizing free energy – while also satisfying data constraints on "sufficient statistics" on $\mathbf{x}$, such that the empirical average is equal to the theoretical average. (This also has a nice connection with Lagrangian duals in convex optimization theory, since both the free energy and entropy are convex functions, and our sufficient statistics are treated as constraints on the resulting problem)

Such an exponential family representation is very convenient for DAG's, because we can write down separable joint distributions as a sum of neighbor-wise interaction energies in our 'Hamiltionian" :

$$P \sim \exp\left(\mathcal{H}(\vec{x})\right) \sim \exp\left(\sum \mathcal{H}_n(x_{n+1}, x_{n-1}, x_n)\right) \tag{602}$$

If we treat our sufficient statistics as simply the conditional mean:

$$T_c(\mathbf{x}) = \mathbf{x} 1_{c=c'} \tag{603}$$

where I've used the $1_{c=c'}$ indicator function above to restrict the mean to each class separately, we get linear discriminant logistic regression, or the softmax function:

$$P(c|\vec{x}) = P(c)P(\vec{x}|c)/P(\vec{x}) \tag{604}$$

$$= \frac{\exp\left(\theta \cdot T_c(\mathbf{x}) - Z(c)\right) h(\mathbf{x}) P(c)}{\sum_{c'} \exp\left(\theta \cdot T_c(\mathbf{x}) - Z(c')\right) h(\mathbf{x}) P(c')} \tag{605}$$

$$= \frac{\exp\left(\theta \cdot T_c(\mathbf{x})\right)}{\sum_{c'} \exp\left(\theta \cdot T_{c'}(\mathbf{x})\right)} \tag{606}$$

As we change the form of our sufficient statistics $T(\vec{x})$, one can see that we change the shape of our decision boundary. This framework is the discriminative one, which has distinct advantages in that it implicitly takes into account interdependence/correlation of the feature $\vec{x}$. Andrew Ng calls Naive Bayes and Logistic Regression a discriminative-generative pair.

16 A simple note on Bias-Variance Decomposition

When one trains a model that is highly flexible, highly "articulate" on a training set, you often get a great training score – be it AUC, accuracy, MSE, etc. But such models – as I've talked about before – often have trouble generalizing to "test" sets, or, the real world. One of the easiest ways to see this is by a simple FOIL operation on the following quantity:

$$Y = f(X) + \epsilon \tag{607}$$

Here X is a random variable – the independent inputs – $f(X)$ is the generative process that creates our object of interest Y – be it cardinal or $\in \mathcal{R}$, and ϵ is drawn from some noise distribution, say a Gaussian process with zero mean, $\mathcal{N}(0, K(X, X'))$. Let $g(X)$ be our picked model for Y. (Normally people write $\hat{f}(X) = g(X)$ but I chose g to avoid confusion.) If we take a look at the mean squared error, we get

$$\begin{aligned} MSE &= \langle |f(X) + \epsilon - g(X)|^2 \rangle & (608)\\ &= \langle f(X)^2 \rangle + \langle \epsilon^2 \rangle + \langle g(X)^2 \rangle - 2\langle \epsilon \rangle \langle f(X) \rangle - 2\langle \epsilon \rangle \langle g(X) \rangle - 2\langle f(X) \rangle \langle g(X) \rangle & (609) \end{aligned}$$

Where I've assumed the noise and our f, g are uncorrelated. We see the terms that are linear in ϵ fall away and we can write:

$$MSE = \langle f(X)^2 \rangle + \langle \epsilon^2 \rangle + \langle g(X)^2 \rangle - 2\langle f(X) \rangle \langle g(X) \rangle \tag{610}$$

Adding and subtracting the mean of our model squared $\langle g(X) \rangle^2$ we get:

$$MSE = \langle (f(X) - \langle g(X) \rangle)^2 \rangle + \mathrm{Var}\,(g(X)) + \langle \epsilon^2 \rangle \tag{611}$$

So now, term by term, we see we have: the squared difference between our data and our average model – the Bias, which quantifies how much our model is "off" (a quantity that will be quite low for complex models and high for simple ones); the variance of the model itself, which quantifies how much our $g(X)$ changes given different training inputs (a quantity that will be high for complex models and low for simple ones) ; and the variance of the noise variable ϵ, which is an ineluctable contribution to our error.

This decomposition illustrates the balancing act we have to play, as model builders, between simplicity and goodness of fit. Refer to this decomposition – or the much harder Chernoff/Union bound from Learning theory – when explaining why your highly un-regularized gradient boosted decision tree – or boosted anything – did a poor job of generalizing! (This doesn't always happen, of course, just a word of caution.)

17 Importance Sampling

18 k means and the EM algorithm

One very popular and very useful algorithm that its taken me a while to get around to is something called the Expectation-Maximization algorithm, or EM. A lot of people treat this thing as a black box, but because I wanted to implement my own handwritten constraints in a K-means and Gaussian mixture model, I had to sit down and read through things.

Turns out the EM algorithm is all about introducing auxiliary variables to your problem – auxiliary data to be exact. Physicists solve things all the time by "enriching" and the integrating out, which is exactly what EM does. Let's say we a sequence of observations $\vec{x}_i$ for $i = 1...N$, and we would like to estimate their density. Problem is, the seem to form some awful distribution, something that would be impossible to model with a Gaussian distribution or a Poisson or any “sane” thing. We could resort to non-parametrics, such as Kernel Density estimation but another idea may be to say “Hey, I bet the first data point came from a Gaussian distribution, but one that was centered over THERE. Let’s call it Gaussian A. And I bet the second was drawn from the same Gaussian A. But the third, which is really far away, was drawn from an entirely different Gaussian distribution, which we can call B . . . ” etc.

What we’re doing here is introducing “membership”, which of course has a close relation to clustering and space segmentation, but more on that later. We now have our auxiliary variables in the problem.

$$\vec{x}_i, \vec{z}_i \tag{612}$$

For each i, $\vec{x}$ is the observable, living in lets say D dimensions, since each data point has D features, while $\vec{z}$ is the membership. For the Gaussian mixture case $\vec{z}$ lives in K dimensions, where K is the number of Gaussians – or clusters – we allow to build the space. Now it’s pretty simple to write down the log Likelihood, but we have to think about what parameters we are conditioning on. We’ve got the mean and variance of EACH Gaussian, let’s call them μ_k, Σ_k and we’ve also got the probability of membership $\vec{z}_i$, for every i, essentially its PDF. Let’s call this $\vec{\phi}$, which note, has K components and is in general a multinomial for the whole data set, multinoulli for a single draw. (The attentive reader will note that I’m taking notation directly from Andrew Ng and Kevin Murphy, who both have great notes on this).

$$\mathcal{L}(X|\phi, \mu_k, \Sigma_k) \;=\; \log\left(P(X|\phi, \mu_k, \Sigma_k)\right) \tag{613}$$

Let’s take this a step at a time. If we assume that every data point $\vec{x}_i$ is independent, then we can write

$$\mathcal{L}(X|\phi, \mu_k, \Sigma_k) \;=\; \sum_i \log\left(P(\vec{x}_i|\phi, \mu_k, \Sigma_k)\right) \tag{614}$$

Now, if we want to include our auxiliary variables, we have to include them explicitly inside the log but marginalize over them:

$$\mathcal{L}(X|\phi, \mu_k, \Sigma_k) \;=\; \sum_i \log\left(\sum_{z_i} P(\vec{x}_i, z_i|\phi, \mu_k, \Sigma_k)\right) \tag{615}$$

And now, we can use a very interesting trick. My first intuition at this point was to expand the joint inside the log like so

$$\mathcal{L}(X|\phi, \mu_k, \Sigma_k) \;=\; \sum_i \log\left(\sum_{z_i} P(\vec{x}_i|\mu_k, \Sigma_k) P(z_i|\phi)\right) \tag{616}$$

and keep working but apparently there’s a more useful way to do things. If we divide and multiply by some UNKNOWN PDF in z_i, $Q(z_i)$ we can use Jensen’s equality to write:

$$\begin{aligned}
\mathcal{L}(X|\phi, \mu_k, \Sigma_k) \;&=\; \sum_i \log\left(\sum_{z_i} Q(z_i)\frac{P(\vec{x}_i, z_i|\phi, \mu_k, \Sigma_k)}{Q(z_i)}\right) && (617)\\
&\geq\; \sum_i \sum_{z_i} Q(z_i) \log\left(\frac{P(\vec{x}_i, z_i|\phi, \mu_k, \Sigma_k)}{Q(z_i)}\right) && (618)
\end{aligned}$$

We can do this because log is a concave function, and for any concave function we have:

$$E\left[f(x)\right] \quad \leq \quad f(E\left[x\right]) \tag{619}$$

Our expectation is over z_i and so now we see that our log loss is strictly greater than this somewhat easier expression on the RHS. But, it kind of looks familiar... its the negative KL divergence between the two distributions in z_i! If we want maximize our lower bound and make it as small as possible, we need to minimize the KL, divergence, essentially setting:

$$P(\vec{x}_i, z_i|\phi, \mu_k, \mathbf{\Sigma}_k) \quad = \quad \mathrm{const}Q(z_i)\ \forall\, z_i \tag{620}$$

So if we sum both sides in z_i we get the marginalized distribution on the left and a constant on the right:

$$\sum_{z_i} P(\vec{x}_i, z_i|\phi, \mu_k, \mathbf{\Sigma}_k) \quad = \quad \sum_{z_i} \mathrm{const}Q(z_i) \tag{621}$$

$$P(\vec{x}_i|\phi, \mu_k, \mathbf{\Sigma}_k) \quad = \quad \mathrm{const} \tag{622}$$

i.e. the best candidate for $Q(z_i)$ is the conditional membership, based on the datapoint x_i itself!

$$Q(z_i) \quad = \quad \frac{P(\vec{x}_i, z_i|\phi, \mu_k, \mathbf{\Sigma}_k)}{P(\vec{x}_i|\phi, \mu_k, \mathbf{\Sigma}_k)} \tag{623}$$

$$= \quad P(z_i|\vec{x}_i, \phi, \mu_k, \mathbf{\Sigma}_k) \tag{624}$$

So now, with this Q in hand in we have a tight lower bound on the log likelihood, which we can then maximize in $\mu_k, \mathbf{\Sigma}_k$:

$$\mathcal{L}(X|\phi, \mu_k, \mathbf{\Sigma}_k) \quad \geq \quad \sum_i \sum_{z_i} P(z_i|\vec{x}_i, \phi, \mu_k, \Sigma_k) \log\left(P(\vec{x}_i|\phi, \mu_k, \mathbf{\Sigma}_k)\right) \tag{625}$$

The EM algorithm simply consists of the following two steps:

1. Given μ_k, Σ_k and the data X, calculate $P(z_i|\vec{x}_i, \phi, \mu_k, \Sigma_k)$. (M-step)
2. Given the above, maximize w/r/t μ_k, Σ_k. (E-Step)
3. repeat

So, we are essentially calculating data point memberships – M step – and then optimizing the log likelihood, as we always would – E step. The difference here is that we don't set hard memberships in our model. We let there be fractional or "soft" memberships in this likelihood expression.

In algorithms like K-means, which actually works on the same EM principle with some drastic assumptions we have hard memberships, and ubuitous covariance matrices across all Gaussians – or clusters – K.

The beautiful thing about EM, and I don't have time to get into it now, is that it's gauranteed to monotonically increase the log likelihood iteration by iteration, and this is simply due to the convexity properties we talked about earlier.

One thing I was hoping to accomplish earlier was to restrict the membership in certain clusters, based on number, or some other characteristic of the data set. This can be accomplished by fiddling with $Q(z_i)$ and having the z_i memberships no longer be independent across the dataset. Could be hard in practice for EM, but for K-means, it simply involves a heuristic of kicking the worst-fitting member out to another cluster. This might involve some hot-potatoe-ing. I'll have to do some checks to see.

19 Facility Location as Adaptive K-Means

In linear programming, a typical problem is the following: we have N facilities, which we can choose to open or not open, in order to serve M customers, who are strewn throughout the world. We would like to minimize the operational cost of serving our customers from these facilities, which can be summarized as:

$$\text{startup costs} = \sum_f s_f y_f \tag{626}$$

$$y_f = 0, 1 \tag{627}$$

Where y_f is a binary variable – zero or one – based on whether we choose to open facility f. s_f is obviously the associated cost for opening that specific facility. We also have transit costs from serving our customers, which might be of the very simple form:

$$\text{transit costs} = \sum_c \sum_f X_{cf} D(\mathbf{x}_c, \mathbf{x}_f) \tag{628}$$

$$X_{cf} = 0, 1 \tag{629}$$

Where X_{cf} is a binary matrix that denotes whether customer c was assigned to facility f. $D(x_1, x_2)$, is our distance metric, which could be Euclidean or ”Manhattan” – where we only take steps to the left or right, up or down, no diagonal lines between destinations – or maybe just a query in google maps.

Adding these to costs together we get a loss function of sorts:

$$J = \sum_c \sum_f X_{cf} D(\mathbf{x}_c, \mathbf{x}_f) + \sum_f s_f y_f \tag{630}$$

which we would like to minimize. There are some simple constraints on our binary variables that we can articulate mathematically. Say each facility has an associated production capacity c_f and each customer has an associated demand d_c. Then we have to make sure that the assignments are not overworking our facilities:

$$\sum_c X_{cf} \leq c_f y_f \, \forall \, f \tag{631}$$

Notice I’ve multiplied by y_f on the RHS. Another constraint we need to have is that every customer is served by exactly one facility – no redundancy:

$$\sum_f X_{cf} = 1 \, \forall \, c \tag{632}$$

We can also require more explicitly that facilities who are turned off do not get any assignments:

$$X_{cf} \leq y_f \quad \forall \, c, f \tag{633}$$

That’s a ton of constraints, but each of them are linear and so we have well defined linear program. Given N facility locations and M customer locations, with the associated capacities, demands, and start up costs, we can find the global optimum of this thing with $N(M + 1)$ decision variables, X and y.

Turns out this problem is very similar to K means. Because what we are essentially doing is breaking down our customers into ”facility clusters”. What if we promoted all customers to facilities themselves? (Or at least gave them the option). Then the distance function $D(\mathbf{x}_1, \mathbf{x}_2)$ is simply the square root of the L_2 norm associated with a mixture of Gaussians with a diagonal covariance matrix. Let $\mathbf{x}_2$ be the facility, or centroid of the k^{th} cluster, then we have:

$$\begin{aligned} \text{Let } \mathbf{x}_2 &= \mu_k && (634)\\ \mathbf{\Sigma} &= \mathbf{1} && (635)\\ \text{Then } D(\mathbf{x}_1, \mathbf{x}_1) &= \sqrt{(\mathbf{x}_1 - \tilde{\mu}_k)\, \mathbf{\Sigma}^{-1} (\mathbf{x}_1 - \tilde{\mu}_k)} && (636) \end{aligned}$$

For a mixture of Gaussians density Estimation, our PDF is:

$$P(\mathbf{x}) = \sum_{k=1}^{K} \mathcal{N}(\mu_k, \mathbf{\Sigma}) \qquad (637)$$

and the log likelihood for our sequence of customers/facilities is:

$$-\log \mathcal{L}(X|\{\mu\}_{k=1}^{K}) = \sum_k \sum_{n \in c_k} \frac{(\mathbf{x}_n - \mu_k)^2}{2\sigma^2} + K\sqrt{2\pi\sigma^2} \qquad (638)$$

I’ve kept the σ^2 terms for clarity. Notice that the second term in this negative log likelihood, or loss function is the normalization factor: it penalizes a high level of total clusters K, preventing overfitting. This loss function – apart from the quadratic nature of the summand – is exactly like the objective in our facility location problem. The startup cost for every ”facility” in this example would be

$$s_f = \sqrt{2\pi\sigma^2} \ \forall f \qquad (639)$$

And the variance of our gaussians we would set to unity: $\sigma \to 1$. Let’s rewrite the negative log likelihood, but in decision variable language:

$$\begin{aligned} J' &= \frac{1}{2} \sum_i \sum_j X_{ij} D(\mathbf{x}_i, \mathbf{x}_j)^2 + \sum_j s_j y_j && (640)\\ s_i &= \sqrt{2\pi} && (641)\\ & && (642) \end{aligned}$$

We have not yet specified the capacity of our clusters or the ”demand” of each data point. A simple choice would be to set some upper limit on the size of each cluster, and give every data point the same weight or demand:

$$\begin{aligned} d_i &= 1 \ \forall\, i && (643)\\ c_j &= c_{\max} \ \forall\, j && (644) \end{aligned}$$

This is still a linear program, just with a slightly different list of coefficients on the first term! To solve such a system we would need $N(N+1)$ decision variables, which could be prohibitive with large systems, but thanks to interior point solvers and very clever routines for integer programming we can actually run this ”adaptive” K-means algorithm in polynomial time!

20 Stratified Sampling

Let's say you want to estimate how "fair" a coin is. What do you do? Probably flip the coin many times and measure the average heads vs. tails rate. In statistical speak, what you're doing is calculating the sample mean:

$$\hat{p} = \sum_{i=1}^{N} \frac{x_i}{N} \tag{645}$$

Which is the maximum likelihood estimate of "heads" probability p. Note that above $x_i = 0, 1$, and so we are just summing up the total number of heads and dividing by the total number of flips. Pretty straightforward. Granted, the true value of p might be different from our estimator $\hat{p}$, and we can characterize this "experiment" variance by the variance of the estimator:

$$\mathrm{Var}(\hat{p}) = \frac{p(1-p)}{N} \tag{646}$$

Looking at the equation above, you can see that as the number of flips $N \to \infty$, we have zero variance, and thus are "infinitely" sure of our probability $\hat{p} = p$. No error bars. This is called the law of large numbers, and is essentially why larger surveys are more robust.

Now that's all well and good, but it turns out you can do a bit better than this simple sample mean, by doing something called stratification. Take for example, K different people, all flipping the same type of coin a total of N times. If we take the sample mean of each person and add them up, we get a different estimator:

$$\hat{p}_{\mathrm{strat}} = \sum_k w_k \hat{p}_k = \sum_k w_k \left(\sum_{i=1}^{n_k} \frac{x_{ik}}{n_k} \right) \tag{647}$$

Where the n_k is the number of coin flips for the k^{th} person, and we require $\sum_k n_k = N$, to compare with our estimator from before. It turns out that if we choose the weights proportional to the number of flips:

$$w_k = \frac{n_k}{N} \tag{648}$$

we get algebraically the same thing as our sample mean above, but the statistical properties are different, because our variance now becomes

$$\mathrm{Var}(\hat{p}_{\mathrm{strat}}) = \sum_k w_k^2 \frac{\sigma_k^2}{n_k} \tag{649}$$

Notice that now we have a weighted sum of in-person variances, which, if we assume to all be the same, we get:

$$\begin{aligned}
\mathrm{Var}(\hat{p}_{\mathrm{strat}}) &= \sum_k w_k^2 \frac{p(1-p)}{n_k} && (650)\\
&= \sum_k \frac{n_k^2}{N^2} \frac{p(1-p)}{n_k} && (651)\\
&= \sum_k \frac{n_k}{N^2} p(1-p) && (652)\\
&= \frac{p(1-p)}{N} && (653)
\end{aligned}$$

The same variance that we got before! So how did this help at all? Well, the key is that if one part of our sample – which in this case is a set of people flipping coins – had a lower variance, we would have a tighter estimate on p. A good way to explain this is to just look carefully at the sum:

$$\mathrm{Var}(\hat{p}_{\text{strat}}) = \sum_k \frac{w_k^2}{n_k} \sigma_k^2 \tag{654}$$

When one of our terms σ_k^2 is very small, we reduce our total variance. There is a tradeoff between in-strata sample size n_k and in-strata variance σ_k^2. To control the total variance of our estimator, we should

where m and c are binary variables. One way to estimate this probability is to take a random sampling of people from the population with a cancer

Where the true value of μ – note that this is different than our estimate $\hat{\mu}$ – is the probability that a single flip yields heads. The variance of our estimate on μ is

$$\mathrm{Var}(\hat{\mu}) = \tag{655}$$

21 Fisher's Exact Test

Let's say you are asked to solve a binary classification problem ($y = 0, 1$) with very few training examples ($N < 1000$) and quite a few, possibly predictive features ($d > 1000$). The standard question of "how the heck do I feature select?" becomes very relevant, and in particular, "how the heck do I feature select with so few training examples?!?".

For Categorical features, one of the best ways to test for significance – i.e. a non-null relationship between a feature column and a label column – is Fisher's exact test and Laplace-smoothed lift.

Fisher's exact test is a combinatoric way of examining a contingency (or pivot) table. Let's say we have two columns x and y, which both take on, in the simplest case, only two values: true and false. If we were to make pivot table, we'd have the number of pair-wise events in a 2x2 grid.

$$\mathrm{pivot}(x, y) = \begin{pmatrix} n_{00} & n_{01} \\ n_{10} & n_{11} \end{pmatrix} \tag{656}$$

Where n_{11} corresponds to the number of times x and y were both True, n_{00} corresponds to the number of times x and y were both false, n_{10} the number times x was true and y false, etc. (This can be done easily in pandas by writing something like *pandas.pivot(dataFrame,index=x,columns=y,aggFunc=len,fillna=0)*.)

We can see that the sum of the entries $n_{11} + n_{01} + n_{10} + n_{00} = N$ equals the number of training examples, and that the sum over rows or columns equals the marginal counts. $n_{11} + n_{10} = R_1$ equals the number times x was true; $n_{01} + n_{11} = C_1$ equals the number times y was true; $n_{01} + n_{00} = R_0$ equals the number times x was false, etc.

What fisher proposed is to take this matrix and ask, given the marginal counts, R_0, R_1, C_0, C_1 – which, if you think about it, correspond to the prior probabilities on x and y: $P(x) = \frac{R_1}{N}, P(y) = \frac{C_1}{N}$ – how likely is the resulting contingency matrix if x and y are independent?

The most naive way to answer that question is to take the prior probabilities on x and y that's given two us by the data:

$$P(x) = p = \frac{R_1}{N} \tag{657}$$

$$P(y) = q = \frac{C_1}{N} \tag{658}$$

and quote our good old multinomial:

$$P(\mathbf{n}) \;=\; \frac{N!}{n_{00}!\; n_{10}!\; n_{11}!\; n_{01}!}\,[pq]^{n_{11}}\,[p(1-q)]^{n_{10}}\,[(1-p)q]^{n_{01}}\,[(1-p)(1-q)]^{n_{00}} \tag{659}$$

which, can be simplified to:

$$P(\mathbf{n}) \;=\; \frac{N!}{n_{00}!\; n_{10}!\; n_{11}!\; n_{01}!}(p)^{n_{10}+n_{11}}(1-p)^{n_{00}+n_{01}}(q)^{n_{11}+n_{01}}(1-q)^{n_{10}+n_{00}} \tag{660}$$

The probability distribution above assumes that there is no relationship between x and y, and so, if we see a contingency table that is very unlikely given the above pdf, we know something is up! But, how sure are we that the prior distribution estimates, p, q are correct? Our sample size N was very small. That's a troubling question, which can be solved by Laplace Smoothing, which sets a uniform prior distribution on $P(x)$ and $P(y)$:

$$P(x) = p \;=\; \frac{R_1+\alpha}{N+\alpha d_x} \tag{661}$$

where d_x is the number of distinct values x can take on – in this case two. And similarly, for y we'd have the prior:

$$P(y) = q \;=\; \frac{C_1+\alpha}{N+\alpha d_y} \tag{662}$$

This helps things a little bit, where α is the hyper parameter between 0 and 1 that controls the "strength" of our uniform prior. But one also might worry if using a multinomial is even appropriate, given, for very few datapoints N, the highly discrete nature of our contingency table.

Fisher's exact test explicitly addresses this discreteness aspect through combinatorics.

Let's recall an experiment where one has a drunken man throw N darts at a dartboard with d cells) the number of different ways in which this drunken dart player can get n_1 darts in the first cell n_2 in the second, n_3 in the third, etc. is:

$$W \;=\; \frac{N!}{n_1!n_2!\cdots n_d!} \tag{663}$$

Taking the log of this combinatoric factor and applying stirling's approximation, we get the Shannon entropy:

$$\log W \;=\; -\sum_{i=1}^{d} p_i \log p_i \tag{664}$$

$$p_i \;=\; \frac{n_i}{N} \tag{665}$$

This is all very interesting because, if one looks at the contingency table above, we need only promote our counts $n_{00}, n_{01}, n_{10}, n_{11}$ to a compound index: n_1, n_2, n_3, n_4 and we get the same formula:

$$W \;=\; \frac{N!}{n_{00}!n_{01}!n_{10}!n_{11}!} \tag{666}$$

This is the number of ways one can get a contingency table with the counts n_{ij}. But, this is NOT a probability. It is simply a multiplicity count of some "state" in phase space, n_{ij}. (You'll see above that it's a normalization factor for the multinomial). If we want to convert this multiplicity count to a probability, we have to be like Kolgomorov and divide by the multiplicity of the entire sample space Ω. After all,

$$P(x \in X) \quad = \quad \frac{|X|}{|\Omega|} \tag{667}$$

Where I'm using bars for "multiplicity", or the count of phase space cells within some region.

For our contingency table, above, we can define precisely what that is: all contingency tables with the marginal sums R_0, R_1 and C_0, C_1. This can be written as compound combinatoric factor:

$$\begin{aligned} |\Omega| &= \begin{pmatrix} N \\ C_0 \end{pmatrix} \begin{pmatrix} N \\ R_0 \end{pmatrix} & (668) \\ &= \frac{N!N!}{R_0!R_1!C_0!C_1!} & (669) \end{aligned}$$

And so we have, doing our division:

$$P(n_{00}, n_{01}, n_{10}, n_{11}|R_1, R_0, C_1, C_0) \quad = \quad \frac{R_0!R_1!C_0!C_1!}{N!n_{00}!n_{01}!n_{10}!n_{11}!} \tag{670}$$

Let the joint event $n_{00}, n_{01}, n_{10}, n_{11}$ be specified by $\mathbf{n}$. Then we can write

$$P(\mathbf{n}|\mathbf{R}, \mathbf{C}) \quad = \quad \frac{R_0!R_1!C_0!C_1!}{N!n_{00}!n_{01}!n_{10}!n_{11}!} \tag{671}$$

or, more generally, for non-binary categorical variables (check it!):

$$P(\mathbf{n}|\mathbf{R}, \mathbf{C}) \quad = \quad \frac{\prod_{i=1}^{d_x} R_i! \prod_{j=1}^{d_y} C_j!}{N! \prod_{i,j} n_{ij}!} \tag{672}$$

This is a very interesting formula, because it gives the precise, discrete probability of seeing some contingency table, conforming to marginal counts $\mathbf{R}$ and $\mathbf{C}$. With a little bit of algebra, one will see that this combinatoric probability converges to the multinomial we quoted above, by noting:

$$\lim_{N \to \infty} N! \quad \approx \quad \left(\frac{N}{e}\right)^N \tag{673}$$

and so we get:

$$P(n_{00}, n_{01}, n_{10}, n_{11}|R_1, R_0, C_1, C_0) \quad \approx \quad \frac{\left(\frac{R_0}{e}\right)^{R_0}\left(\frac{R_1}{e}\right)^{R_1}\left(\frac{C_0}{e}\right)^{C_0}\left(\frac{C_1}{e}\right)^{C_1}}{(N/e)^N\left(\frac{n_{00}}{e}\right)^{n_{00}}\left(\frac{n_{01}}{e}\right)^{n_{01}}\left(\frac{n_{10}}{e}\right)^{n_{10}}\left(\frac{n_{11}}{e}\right)^{n_{11}}} \tag{674}$$

All the factors of e cancel out, and we can simplify to get:

$$
\begin{aligned}
pq &= \frac{n_{11}}{N} && (675)\\
p(1-q) &= \frac{n_{10}}{N} && (676)\\
(1-p)q &= \frac{n_{01}}{N} && (677)\\
(1-p)(1-q) &= \frac{n_{00}}{N} && (678)\\
P(\mathbf{n}|\mathbf{R},\mathbf{C}) &= \frac{N!}{\prod_{i,j} n_{ij}!}(\frac{R_0}{N})^{R_0}(\frac{R_1}{N})^{R_1}(\frac{C_0}{N})^{C_0}(\frac{C_0}{N})^{C_0} && (679)\\
&= \frac{N!}{n_{00}!\, n_{10}!\, n_{11}!\, n_{01}!}(p)^{n_{10}+n_{11}}(1-p)^{n_{00}+n_{01}}(q)^{n_{11}+n_{01}}(1-q)^{n_{10}+n_{00}} && (680)
\end{aligned}
$$

The same multinomial formula we found above!

This really isn't so surprising, as it says the combinatoric probability converges to the multinomial with fully continuous priors p, q in the large sample $N \to \infty$ limit, but it is interesting to note.

Now, Fisher, when quoting p-values, or significance tests for a relationship between x, y, would simply use the count of contingency tables that had table counts $\mathbf{n}$ more extreme than what's observed. For instance, let's say we observe a True/True x, y occurence that is higher than expected under the marginals: $n_{11} > Npq$ or $n_{11} > R_1C_1/N$: what's the sum of the probabilities of tables that have an even *higher* n_{11}? This is called a one-tailed pvalue significance test, and for the fisher exact test and the multinomial method, corresponds to a simple sum.

I won't get too into the details of implementation now, but suffice to say, scipy's got a fisher test calculation all on its own!

Now what hasn't been mentioned is lift. And it relates directly to the laplace smoothed priors discussed earlier. Lift is simply:

$$
l(x|y) = \frac{P(x|y)}{P(x)} = \frac{P(x,y)}{P(x)P(y)} \qquad (681)
$$

or, the probability of x taking on some value given y relative to x occurring independently. Lift is a number between zero and infinity, and basically means: how many more times likely is x going to occur given y? For low sample size $N < 1000$, it's probably a good idea to smooth the priors $P(x), P(y)$ giving us:

$$
\begin{aligned}
l(x|y) &= \frac{P(x,y)}{P(x)P(y)} && (682)\\
&= \frac{(N+\alpha d_x)(N+\alpha d_y)}{N}\frac{n_{xy}}{n_x n_y} && (683)
\end{aligned}
$$

Where n_x, n_y is the event count of x and y. n_{xy} is th joint event count of x, y.

22 Restricted multinoulli samples

I was asked an interesting friend the other day, about density estimation. Let's say we're interested in some categorical variable X that can take on values from 1 to C. Then we can write down the PDF of a single sample or "trial" as a multinoulli:

$$P(X) = \prod_{c=1}^{C} \theta_c^{\mathbf{1}_{x=c}} \tag{684}$$

Where $\mathbf{1}$ is the indicator function. If we have many realizations of this random variable X, we get a multinomial, and can write down our pdf in terms of the total number of times X was equal to c as n_c. If we have N observations, that means:

$$P(\vec{n}|\vec{\theta}) = \frac{N!}{n_1!n_2!\cdots n_C!}\theta_1^{n_1}\theta_2^{n_2}\cdots\theta_C^{n_C} \tag{685}$$

where we require

$$\sum_c n_c = N \tag{686}$$

$$\sum_c \theta_c = 1 \tag{687}$$

Note I've used the shorthand $\vec{n} = n_1, n_2, \ldots n_C$ and $\vec{\theta} = \theta_1, \theta_2, \ldots \theta_C$. What we've written above is the likelihood of a sequence of multinoulli observations, given some categorical probability distribution $\vec{\theta}$. One might ask, what's the maximum likelihood estimate of $\vec{\theta}$? If we maximize our likelihood subject to our normalization constraint on the PDF, we have

$$\text{maximize} \qquad P(\vec{n}|\vec{\theta}) = \frac{N!}{\prod_c n_c!}\prod_c \theta_c^{n_c} \tag{688}$$

$$\text{subject to} \qquad \sum_c \theta_c = 1 \tag{689}$$

Taking the log and taking a derivative with respect to θ_c we get:

$$\frac{\partial}{\partial\theta_c}\left(\log P(\vec{n}|\vec{\theta}) + \lambda(1-\sum_c \theta_c)\right) = 0 \tag{690}$$

$$\frac{\partial}{\partial\theta_c}\left(\log N! - \sum_c \log n_c! + \sum_c n_c \log\theta_c + \lambda(1-\sum_c \theta_c)\right) = 0 \tag{691}$$

$$\frac{n_c}{\theta_c} - \lambda = 0 \tag{692}$$

$$\theta_c = \frac{n_c}{\lambda} \tag{693}$$

We can determine λ by summing over c:

$$\sum_c \theta_c = \sum_c \frac{n_c}{\lambda} \tag{694}$$

$$1 = \frac{N}{\lambda} \tag{695}$$

$$\Rightarrow N = \lambda \tag{696}$$

So finally, we have

$$\theta_c = \frac{n_c}{N} \tag{697}$$

Ok, that's all well and good, in line with our intuition. We expect X to fall into the categorical bin c the number of times we observed it in there, normalized by our total observations!

———————————————-

Now, what if during these multinoulli trials, not all of the categorical variables – or not all of our sample space $\Omega = \{1, ...C\}$ – was available? Then we'd have to change our Likelihood estimation, because a pure count isn't quite the right thing to do. Let's nail down some notation. Our data is a sequence of categorical observations

$$\text{Data} = \{x_t\}_{t=1}^T \tag{698}$$

where, at each instance t, our categorical variable comes from a subset of our sample space:

$$x_t \in \Omega_t \subset \Omega \tag{699}$$

This meas, if we were to write down the log likelihood, we'd have something like (assuming each x_t is independent):

$$\log P(\{x_t\}_{t=1}^T|\vec{\theta}) = \sum_t \log P(x_t \in \Omega_t|\vec{\theta}) \tag{700}$$

The probability inside the sum can be written as a marginalization over the bins c not included in the sample set Ω_t

$$P(x_t \in \Omega_t|\vec{\theta}) = \sum_{x_t \not\in \Omega_t} P(x_t|\vec{\theta}) \tag{701}$$

and so our likelihood becomes:

$$\log P(\{x_t\}_{t=1}^T|\vec{\theta}) = \sum_t \log\left(\sum_{x_t \not\in \Omega_t} P(x_t|\vec{\theta})\right) \tag{702}$$

You might at the inner sum and say that all is lost, but if we group the outer sum in terms of common sample set, we can write:

$$\log P(\{x_t\}_{t=1}^T|\vec{\theta}) = \sum_{S\subset\Omega}\left[\sum_{S=\Omega_t} \log\left(\sum_{x_t \not\in \Omega_t} P(x_t|\vec{\theta})\right)\right] \tag{703}$$

and now we see the term in brackets is just another multinomial with sample space Ω_t. Converting our observed variables x_t to counts, given sample space S, we get:

$$\log P(\{x_t\}_{t=1}^T|\vec{\theta}) = \sum_{S\subset\Omega} \log P(\vec{n}_S|\vec{\theta}) \tag{704}$$

You might be worried that we're summing over all possible subsets S, but we'll get to that implementation detail in a moment. The point is, we can now write our Likelihood as:

$$\log P(\{x_t\}_{t=1}^T|\vec{\theta}) = \sum_{S\subset\Omega}\left(\log(N_S!) - \sum_c \log(n_{c,S}!) + \sum_{c\in S} n_{c,S}\log\theta_c\right) \tag{705}$$

To clarify N_S is the total number of observations we drew from the subset S, and $n_{c,S}$ is the count of categorical variable c in all instances of the subset S. Adding our lagrange multipliers once again – one for each subset – we get:

$$\begin{aligned} \text{minimize} \quad & \sum_{S\subset\Omega}\left(\log(N_S!) - \sum_c \log(n_{c,S}!) + \sum_{c\in S} n_{c,S}\log\theta_c\right) && (706)\\ \text{subject to} \quad & \sum_{c\in S}\theta_c = 1\;\forall S && (707)\end{aligned}$$

Taking the derivative once again with respect to θ_c we get a bunch of kronecker deltas – or depending upon how you look at it, indicator functions – in the sums:

$$\begin{aligned} \frac{\partial}{\partial\theta_c}\left(\sum_{S\subset\Omega}\left(\log(N_S!) - \sum_c \log(n_{c,S}!) + \sum_{c\in S} n_{c,S}\log\theta_c\right) + \sum_S \lambda_S(1-\sum_{c\in S}\theta_c)\right) &= 0 && (708)\\ \sum_{S\subset\Omega}\mathbf{1}_{c\in S}\frac{n_{c,S}}{\theta_c} - \sum_{S\subset\Omega}\lambda_S\mathbf{1}_{c\in S} &= 0 && (709)\end{aligned}$$

So we have:

$$\begin{aligned} \sum_{S\subset\Omega}\mathbf{1}_{c\in S}\frac{n_{c,S}}{\theta_c} &= \sum_{S\subset\Omega}\lambda_S\mathbf{1}_{c\in S} && (710)\\ \theta_c &= \frac{\sum_S n_{n_c,S}\mathbf{1}_{c\in S}}{\sum_S N_S\mathbf{1}_{c\in S}} && (711)\end{aligned}$$

or more simply,

$$\theta_c = \frac{n_c}{\sum_S N_S\mathbf{1}_{c\in S}} \tag{712}$$

So what does this final formula mean? It means our best estimate for the probability of our categorical bin c, given the data, is equal to the number of times we saw $X = c$, in general, divided by a new numerator, which, instead of N is the number of times X HAD A CHANCE to be equal to c. Pretty intuitive, but difficult to prove!

Now back to that comment about summing over all subsets. In reality, for most data, you're only going to see a finite number of subsets of your categorical sample space (although things may become a pain if C is really large). The best way to estimate these parameters is to do a masked sum. Let the data matrix be a binarized thing:

$$\begin{aligned} X_{tc} &= 0,1 && (713)\\ \text{where} \quad & t = 1,\dots T, c = 1,\dots C && (714)\end{aligned}$$

where X_{tc} is unity if the t^{th} observation is equal to – or falls into bin – c. Now let our mask matrix be M_{tc}, which is zero if c is not in Ω_t ($c \notin \Omega_t$) and one if it is. Then our estimates I wrote above are:

$$\theta_c = \sum_t X_{tc} M_{tc} \tag{715}$$

which can be done in numpy pretty darn fast.

23 Label Propagation and Semi-Supervised Learning: Gaussian Random Field Method

So, recently I've been reading up on label propagation in semi-supervised learning, which is when you have a great deal of data, but most of it is unlabeled. To put some notation on things, lets say way have a set L:

$$L : \{\mathbf{x}, y\}_{n=1}^{L} \tag{716}$$

which is a set of pairs of input vectors x and output labels y, be they scalar or categorical. And then we have a huge unlabeled set:

$$U : \{\mathbf{x}, \_\_\}_{n=1}^{U} \tag{717}$$

which we would like to infer on. Normally, this use case is motivated when the unlabeled set is much, much larger, $|L| << |U|$. If we are talking about classification, one way to view this problem is through clustering. If we assume that close vectors $\mathbf{x}$, under some metric, have close labels y, we that might motivate a loss function of the form:

$$E(\{y\}) = \sum_{i,j\neq i} W_{ij}(y_i - y_j)^2 \tag{718}$$

Where, we're summing over all pairs of data points i, j, and weighting their difference in label with the matrix W_{ij}. For sanity's sake, W_{ij} should be large when x_i, x_j are close. So W_{ij} goes like one over distance between i, j. For example:

$$E(\{y\}) = \sum_{i,j\neq i} W_{ij}(y_i - y_j)^2 \tag{719}$$

$$W_{ij} = e^{-|x_i - x_j|^2/2\sigma^2} \tag{720}$$

This weighting matrix is simply a function of the euclidean metric, and actually reminds one of an RBF kernel or covariance function... Is there a connection here?

Absolutely.

If we frame our clustering/labeling problem as trying to minimize this loss function, or energy E, it means we can actually frame the likelihood of the labels with a boltzman distribution:

$$P(\{y\}) = \frac{1}{Z} e^{-\sum_{i,j\neq i} W_{ij}(y_i - y_j)^2} \tag{721}$$

(Where Z is the partition function, summing over all configurations of labels). This is extremely interesting, because if you do a little matrix algebra on our energy, we you find that one can re-write the loss as:

$$E = \sum_{i,j\neq i} W_{ij}(y_i - y_j)^2 = \sum_{i,j\neq i} y_i(D_{ij} - W_{ij})y_j \tag{722}$$

$$= \frac{1}{2} y_i L_{ij} y_j \tag{723}$$

$$D_{ii} = \sum_{j'} W_{ij} \tag{724}$$

$$L_{ij} = \mathbf{D} - \mathbf{W} \tag{725}$$

The matrix L_{ij}, above is actually a close cousin of the laplacian operator ∇^2, but we have embedded things in a high-dimensional space because of exponentiation. Notice that our likelihood on the configuration of labels now looks exactly like a Gaussian random field:

$$P(\{y\}) = \frac{1}{Z} e^{-y_i L_{ij} y_j/2} \tag{726}$$

such that $\langle y_i \rangle = 0$ and $\langle y_i y_j \rangle_c = L_{ij}^{-1}$. This discrete pdf on labels is precisely the same as if we had made everything continuous from the get-go:

$$E[y(x)] = \frac{1}{2} \int dx dx' y(x) y(x') K(x, x') \tag{727}$$

$$K(x, x') = e^{-|x-x'|^2/2\sigma^2} \tag{728}$$

$$P(y(x)) = \frac{1}{Z} e^{-\frac{1}{2}\int dx dx' y(x) y(x') K(x,x')} \tag{729}$$

which is a Gaussian random field on the labels, $y(x)$, imposing an RBF correlation function between points x. When integrate the lagrangian in by parts we would get the continuous equivalent of L_{ij}, which is essentially ∇^2 in some new space.

So why do we care about all of this? Well, it turns out that the algorithms people use to propagate labels work exactly like the Helmholtz equation. For instance, one of the easiest things you can do given labeled examples L, is to propagate or "flow" the y's to unlabeled points by the following procedure:

$$y_{t+1} = \mathbf{D}^{-1}\mathbf{W} y_t \tag{730}$$

which, is the same as the helmholtz equation:

$$\left(\frac{\partial}{\partial t} + \nabla^2\right) y_t = 0 \tag{731}$$

$$y_{t+1} - y_t = \nabla^2 y_t \tag{732}$$

$$y_{t+1} = \left(1 - \nabla^2\right) y_t \tag{733}$$

and now note, if we replace ∇^2 with $1 - D^{-1/2}\mathbf{W}D^{-1/2}$, we get

$$y_{t+1} = \mathbf{D}^{-1}\mathbf{W} y_t \tag{734}$$

This is PRECISELY the update scheme – in discrete form – of Helmholtz dynamics. (Although we are doing things not in euclidean space but somewhere else, do to our choice of metric. Because for instance, we could have chosen W_{ij} to be whatever we liked, as long as it goes inverse with distance. Let $g(x_i, x_j)$ be some metric, then we have more generally:

$$W_{ij} = f\left[g(x_i, x_j)\right] \tag{735}$$

and so g defines the space in which we're taking derivatives. Things don't have to be euclidean!

23.1 Markov Random Walks

Szummer and Jaakola (2002) used the exact same frame work written in the last post to propagate labels outwards from a training set via some distance measure. But, they used a Markov random walk, with transition matrix:

$$\begin{aligned} p_{ij} &= \frac{W_{ij}}{\sum_{ik} W_{ik}} \qquad (736) \\ &= \mathbf{D}^{-1}\mathbf{W} \qquad (737) \end{aligned}$$

Notice, this is exactly the same as our transition matrix from before. The best way to view p_{ij} is a conditional probability that a particle lives at position i, given that it was at position j the moment before:

$$p_{ij} = P(x_{t+1} = \mathbf{x}_i | x_t = \mathbf{x}_j) \tag{738}$$

Now, this is only a single step. By markov property we can extend to any number of steps in time t:

$$\begin{aligned} p_{ij}^{(2)} &= P(x_{t+2} = \mathbf{x}_i | x_t = \mathbf{x}_j) \qquad (739) \\ &= \sum_{x_{t+1}} P(x_{t+2} = \mathbf{x}_i | x_{t+1} = \mathbf{x}_j) P(x_{t+1} = \mathbf{x}_i | x_t = \mathbf{x}_j) \qquad (740) \end{aligned}$$

or, in matrix form:

$$\begin{aligned} p_{ij}^{(2)} &= \sum_k p_{ik} p_{kj} = \mathbf{p}^2 \qquad (741) \\ p_{ij}^{(t)} &= \mathbf{p}^t \qquad (742) \end{aligned}$$

this type of framework allows for traversal of particles through our "graph", consisting of the labeled and unlabeled datapoints. It is precisely the same as the Helmholtz algorithm given before, but instead of soft labels $y(x)$ being propagated we have representative particles.

24 Entity Resolution

So, along with label propagation, I've also been thinking about entity resolution, which is basically the same thing if you frame the problem correctly.

Let's say you have a set of all labeled data:

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N \tag{743}$$

Where y_i can be a class – zero or one – as we were talking about earlier, or a unique ID. What we would like to do is compare pair-wise our datapoints and see if the y_i, y_j's are equal. This means that every pair is a probability instance, we 'd like to assign them a "two-peas-in-a-pod" probability. One way of doing this is with our similarity matrix, mentioned before:

$$P(y_i = y_j|x_i, x_j) = \mathbf{W}_{ij} = e^{-\alpha_n g_n(x_i,x_j)} \tag{744}$$

Where in the exponent we have a linear combination of metrics. They can be Euclidean, Minkowski, cosine, what have you – each with a weight α_n. (This is particularly useful with string comparison, as some metrics are more informative of others). We can also use simple logistic regression:

$$P(y_i = y_j|x_i, x_j) = \sigma\left(-\mathbf{W}_{ij}\right) = \frac{1}{1 + e^{\alpha_n g_n(x_i,x_j)}} \tag{745}$$

(it turns out that this probability is flipped the wrong way if we kept the negative sign in the exponent, which can be seen by a simple plot). If we want to learn the optimal α_n's we can use gradient descent on some specified objective function. The graph based formulation is motivated by "Hidden Markov Random Field" which penalizes different labels between close points – as specified by g.

$$\begin{aligned} E &= \sum_{i,j\neq i} W_{ij}(y_i - y_j)^2 = \sum_{i,j\neq i} e^{-\alpha_n g_n(x_i,x_j)}(y_i - y_j)^2 && (746)\\ &= \sum_{i,j\neq i} P(y_i = y_j)(y_i - y_j)^2 && (747)\\ E &= \mathcal{E}\left((y_i - y_j)^2\right) && (748) \end{aligned}$$

we see that this energy E is just the expectation value of the pairwise label distance, a certain type of empirical Risk! (E can also be treated as the log loss or negative log probability of the configuration $\{y\}$).

Similarly, for logistic regression we just have our log loss. Both objective functions are differentiable with respect to the metric weights α_n, so if we want to LEARN what comparators between x_i, x_j are important, we simply use gradient descent on our labeled examples! To extend labels/matches to unlabeled points, we use the pairwise probabilities specified above.

25 Estimating the Survival Function

In survival analysis, the key quantity of interest is something called the survival function, $S(t)$, which is the probability that I'm going to live, *at least* as long as I've lived already:

$$S(t) = P(T \geq t) \tag{749}$$

along with something called the hazard function, which is the probability that I *die* today, at time t, given that I've lived up until now:

$$\lambda(t) = P(T = t | T \geq t) = P(t)/S(t) \tag{750}$$

This hazard is a conditional probability, and comes about because survival analysis and survival-like problems are implicitly *sequential.*

When estimating $S(t)$ from data, one often uses the Kaplan Meier Estimator, which is a cumulative product of the number of people who "died" at time t, d_t, and the number of people who were "alive" at time t, n_t:

$$\hat{S}(t) = \prod_{t'<t} \left(1 - \frac{d_{t'}}{n_{t'}}\right) \tag{751}$$

This is actually just a cumulative product of time-step survival probabilities, or one minus the hazard:

$$\begin{aligned} \hat{S}(t) &= \prod_{t'<t} (1 - \lambda_{t'}) && (752) \\ &= \prod_{t'<t} p_{t'} && (753) \end{aligned}$$

If we were to ask ourselves, "what's the variance of this estimator?", we'd have to use some fancy tricks. The first of which is noticing that we don't have good ways of combining variances in a **product**, but we do have good ways of combining variance for **sums**. So let's take the log transform of our estimator:

$$\begin{aligned} \log\left(\hat{S}(t)\right) &= \sum_{t'<t} \log(1 - \lambda_{t'}) && (754) \\ &= \sum_{t'<t} \log(p_{t'}) && (755) \end{aligned}$$

And note that, the variance of the log can be computed by a simple taylor expandsion of a random variable about its mean:

$$\begin{aligned} X &\sim ? && (756) \\ \langle X \rangle &= \mu && (757) \\ \mathbf{Var}(X) &= \sigma^2 && (758) \\ \log(X) &\approx \mu + \frac{X - \mu}{\mu} + O((X-\mu)^2) + \dots && (759) \\ \mathbf{Var}\left(\log(X)\right) &= 0 + \frac{\mathbf{Var}(X)}{\mu^2} && (760) \\ &= \frac{\sigma^2}{\mu^2} && (761) \end{aligned}$$

So we have:

$$\begin{aligned}\log\left(\hat{S}(t)\right) &= \sum_{t'<t} \log\left(1-\lambda_{t'}\right) && (762)\\ &= \sum_{t'<t} \log(p_{t'}) = \frac{1}{\hat{S}(t)^2}\mathbf{Var}(\hat{S}(t)) && (763)\end{aligned}$$

Using this transform on our formula above, we have

$$\mathbf{Var}\left(\hat{S}(t)\right) = \hat{S}(t)^2 \mathrm{Var}\left(\sum_{t'<t} \log(p_{t'})\right) \quad (764)$$

Luckily, if we assume independence, the variance of the sum is the sum of the variances, so we can treat each p_t as an independent binomial draw, with variance $p_t(1-p_t)/n_t$, where n_t is the "sample size" of our survival curve at time t.

Working through some nasty algebra, and another use of the variance of the log identity we get:

$$\begin{aligned}\mathbf{Var}\left(\hat{S}(t)\right) &= \hat{S}(t)^2 \sum_{t'<t} \mathrm{Var}\left(\log(p_{t'})\right) && (765)\\ &= \hat{S}(t)^2 \sum_{t'<t} \frac{1}{p_{t'}^2}\mathrm{Var}\left(p_{t'}\right) && (766)\\ &= \hat{S}(t)^2 \sum_{t'<t} \frac{p_{t'}}{n_{t'}(1-p_{t'})} && (767)\end{aligned}$$

We see that variance of the estimator goes like the cumulative sum of one over the sample size at each time t:

$$\mathbf{Var}\left(\hat{S}(t)\right) \sim \sum_{t'<t} \frac{1}{n_{t'}} \quad (768)$$

Now, when dealing with very large data, say billions of survival events, it can be difficult to get these death counts as a function of time, due to a few implementation details, and the resistance of cumulative sums to parallelization. So, what people often do, is the estimate the survival curve at multiple snapshots, M, and then take the average of the snapshot estimates:

$$\begin{aligned}\hat{S}_M(t) &= \sum_{m=1}^{M} \frac{\hat{S}_m(t)}{M} && (769)\\ &= \sum_{m=1}^{M} \frac{\prod_{t'<t}\left(1-\lambda_{mt'}\right)}{M} && (770)\end{aligned}$$

This estimator will have the same mean as our "full history" estimator, but slightly different variance properties. As we know, the variance of a mean goes like one over the sample size:

$$\mathbf{Var}\left(S_M(t)\right) = \frac{1}{M}\mathbf{Var}\left(S_m(t)\right) \quad (771)$$

But what's the variance of each snapshot estimate? Simply our old formula, with the population count n_{mt} instead of n_t. Or, in english, the number of people who were "alive" at time t in snapshot m, rather than the total number of people who were alive at time t. Strictly, $n_{mt} < n_t$. If we assume our snapshots are evenly populated with "alive" people at each time t, we will have $n_{mt}M \approx n_t$.

And so, comparing the variance of our estimators, we see:

$$\begin{aligned} \mathbf{Var}\left(S(t)\right) &= \hat{S}(t)^2 \sum_{t'<t} \frac{p_{t'}}{n_{t'}(1-p_{t'})} && (772) \\ \mathbf{Var}\left(S_M(t)\right) &= \frac{1}{M}\hat{S}_M(t)^2 \sum_{t'<t} \frac{p_{mt'}}{n_{mt'}(1-p_{mt'})} && (773) \end{aligned}$$

Taking the ratio of the variances, we get, since the means are equal ($\hat{S}(t)^2 = \hat{S}_M(t)^2$):

$$\frac{\mathbf{Var}\left(S_M(t)\right)}{\mathbf{Var}\left(S(t)\right)} = \frac{\sum_{t'<t} \frac{p_{mt'}}{n_{mt'}(1-p_{mt'})}}{\sum_{t'<t} \frac{p_{t'}}{n_{t'}(1-p_{t'})}} \tag{774}$$

Assuming equal sample size across snapshots, we can make the replacement, $n_{mt} = n_t/M$:

$$\frac{\mathbf{Var}\left(S_M(t)\right)}{\mathbf{Var}\left(S(t)\right)} \approx \frac{\sum_{t'<t} \frac{p_{mt'}}{n_{t'}(1-p_{mt'})}}{\sum_{t'<t} \frac{p_{t'}}{n_{t'}(1-p_{t'})}} \tag{775}$$

And, assuming the $p_{mt} \approx p_t \forall t$, we get the very simple ratio:

$$\frac{\mathbf{Var}\left(S_M(t)\right)}{\mathbf{Var}\left(S(t)\right)} \approx 1 \tag{776}$$

How well does this mean we're doing? Well, it means that the variances of both methods are comparable. Which is surprising! If we want to probe deeper, and understand whether or not there is a difference between the two sampling strategies, we would have to closely inspect the cumulative sum:

$$\sum_{t'<t} \frac{p_{mt'}}{n_{mt'}(1-p_{mt'})} \sim \sum_{t'<t} \frac{p_{t'}}{n_{t'}(1-p_{t'})} \tag{777}$$

26 LARS: Least Angle Regression

Typically, when one performs a multivariate regression on y with features $\mathbf{x}$, one uses the normal equation:

$$\begin{aligned} \text{Data} &= (X_{nj}, Y_n) && (778)\\ \hat{y}_n &= \theta_j X_{nj} + \epsilon && (779)\\ \epsilon &\sim N(0, \sigma_n^2) && (780)\\ \theta_k &= (X_{nk} X_{nj})^{-1} X_{nj} Y_n && (781)\\ &= (\mathrm{Cov}(\mathbf{x}_k, \mathbf{x}_j))^{-1} \,\mathrm{Cov}(\mathbf{x}_j, y) && (782) \end{aligned}$$

This is all well and good, for estimating regression coefficients in a multivariate model, but one often wants to aggressively feature select, or search for only those inputs that are most relevant to prediction. One approach to this is penalizing the likelihood with an L1 or L2 prior, which results in a slightly different normal equation:

$$\theta_k = (X_{nk} X_{nj} + \epsilon)^{-1} X_{nj} Y_n \qquad (783)$$

Where ϵ exerts downward force on the regression coefficients, through a Gaussian prior. Beyond this, one can also recursively regress on the *residuals*, which is how least angle regression works.

Let's say we would like to first construct a model with a *single* input for prediction: a linear model would of course choose the feature – and the coefficient – that have the highest correlation with the label:

$$\begin{aligned} f_1(\mathbf{x}_n) &\approx y_n = \theta_1 x_1 + \epsilon && (784)\\ \theta_1 &= \frac{\rho_1}{\sigma_1^2} && (785)\\ \rho_1 &= \mathrm{Cov}(x_1, y) && (786)\\ \sigma_1^2 &= \mathrm{Cov}(x_1, x_1) && (787) \end{aligned}$$

Next, we would like to add a new feature to our model, but instead of regressing on the label itself with x_1 and x_2, we are going to find the coefficient that is most correlated with the residual r from our first model:

$$\begin{aligned} r_n &= (y_n - \theta_1 x_1) && (788)\\ r_n &\approx \theta_2 x_2 + \epsilon && (789)\\ \theta_2 &= \frac{\rho_2}{\sigma_2^2} && (790)\\ \rho_2 &= \mathrm{Cov}(x_2, r_n) && (791)\\ \sigma_2^2 &= \mathrm{Cov}(x_2, x_2) && (792) \end{aligned}$$

27 Colinearity

27.1 Part 1

I've heard a lot comments – some captions, some cowardly – about ”co-linearity” recently, from both colleagues at work and friends using statistics in their jobs. And, well, GUESS WHAT? Co-linearity is not as scary as it used to be! Many people don't realize that there are a variety of ways to avoid or control ”co-linearity” in data when performing basic regressions, and I want to take some time to outline them.

Let's begin by saying we've got regression problem on our hands: one where we have N examples of some feature vectors $\mathbf{x}$, of dimension D, contained in an $N \times D$ data matrix:

$$X = \begin{pmatrix} \leftarrow \vec{x_1} \rightarrow \\ \leftarrow \vec{x_2} \rightarrow \\ \vdots \\ \leftarrow \vec{x_N} \rightarrow \end{pmatrix} \tag{793}$$

and, N real-valued response variable examples, Y:

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \tag{794}$$

Our goal is to write a linear model of some sort:

$$\hat{y_n} \approx \beta \cdot \mathbf{x_n} + \epsilon \tag{795}$$

Where we assume the errors are drawn from some probability distribution. In standard regression problems Normal, so we'll keep it that way:

$$\epsilon \sim \mathcal{N}(0, \sigma^2) \tag{796}$$

Now, as we've covered before in this blog, the likelihood of the data – or, the co-occurence of features and labels we see in the world (X, Y) – given some model, specified by $\mathtt{beta}$, is equal to:

$$P(X, Y|\beta) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-(y_n - \beta \cdot x_n)^2/2\sigma^2\right) \tag{797}$$

Where, by taking a product above, we assume each data instance $n = 1, \ldots N$ is independent. Taking the log of this likelihood we get:

$$\mathcal{L}(X, Y|\beta) = -\frac{N}{2}\log(2\pi\sigma^2) - \sum_{n=1}^{N} \frac{(y_n - \beta \cdot x_n)^2}{2\sigma^2} \tag{798}$$

This is a convex function of β, meaning that if we set the derivative equal to zero, we are guaranteed to find a global maximum / minimum (very good), and so our MLE or maximum likelihood estimate of the model β becomes, if we write things now in matrix notation:

$$\mathcal{L}(X, Y|\beta) = -\frac{N}{2}\log(2\pi\sigma^2) - (\beta_d X_{nd} - Y_n)^2 2\sigma^2 \tag{799}$$

Taking the derivative with respect to β_d (a gradient) we get:

$$\begin{aligned} \frac{\partial \mathcal{L}(X, Y|\beta)}{\partial \beta_d}|_{\beta=\beta'} &= (\beta_l X_{nl} - Y_n) X_{nd} = 0 && (800) \\ \beta_l X_{nl} X_{nd} &= X_{nd} Y_n && (801) \\ \beta_l &= (X_{nl} X_{nd})^{-1} X_{nd} Y_n && (802) \end{aligned}$$

This is called the Normal Equation, can actually be well understood that noting:

$$X_{nl}X_{nd} \approx N\text{Cov}(x_l, x_d) \tag{803}$$

$$X_{nl}Y_n \approx N\text{Cov}(x_l, y) \tag{804}$$

which, words, is the covariance betwen the l^{th} d^{th} components of the feature vector and the covariance between the l^{th} feature and the target, y.

There's a very important thing to notice here, straight off the bat, which is that the Normal Equation – which is the standard way of solving regression, or OLS (ordinary least squares) problems – accounts for interactions between the features: we can see it in the "discounting" factor of the inverted matrix, above. Highly correlated features will dampen each other's effect, which is very, very cool. Regression coefficients β_d represent the "net" effect of the d feature, not the "gross" effect, as one would get by doing a single, univariate regression of x_d on y. This is important to keep in mind, but we're not out of the – or even into the – co-linearity woods yet.

————————————-

People get upset or concerned about colinearity when they want:

1. Interpretable Models
2. Stable Regression Coefficients β in the face of changing data.
3. A bone to pick with a model or feature set that they don't trust or understand.

Now, most of the "spookiness" of co-linearity comes from linear algebra, and **the complete absence of Bayesian Statistics in Traditional circles of past Statisticians, where putting priors on regression coefficients is equivalent to regularizing, and therefore controlling and containing, co-linearity.**

Take for example a data matrix where we have $N = 15$ data points in our set, but $D = 45$ features. The old-time statisticians might tell you that the problem is ill-specified or ill-defined, because if we create a regression model with 45 degrees of freedom, there simply aren't enough data points to "figure out" what's going on. And that's true, but it really comes from the fact that when inverting a matrix – as we're doing above – with linearly dependent columns, we could run into a lot of numerical trouble.

This has to be the situation in the case I just mentioned above, as it is impossible for the square matrix $X_{nl}X_{nd}$ – which is $D \times D$ – to be properly invertible. And this is simply because the column space of X is at most of dimension $N = 15$. I won't get into the dirty details, but suffice it to say that when you tried to regress in such a situation, solving using the old methods, you were hosed.

The way to wiggle out of it, was to sub-select the features, such that $D < N$, and move on with the analysis. One could also use PCA to "represent" the matrix $X_{nl}X_{nd}$ in terms of its most prominent eigenvectors, but in modern times this is a brute way to do things, and especially with genetic data, where the feature space is far more dimensional than the number of examples we have (by a factor of 10, 100 or even a 1000), we can do better than that.

27.2 Part 2

Last post was concerned with co-linearity in regression problems, and how one chooses to deal with it. The Normal equation and was mentioned before:

$$\beta_d = (X_{nd}X_{nl})^{-1} X_{ml}Y_m \tag{805}$$

and, we can also introduce the "hat" matrix:

$$\hat{Y}_n = X_{nd}\beta_d \tag{806}$$
$$= X_{nd}\left(X_{n'd}X_{n'l}\right)^{-1}X_{ml}Y_m \tag{807}$$
$$\mathbf{H}_{nm} = X_{nd}\left(X_{n'd}X_{n'l}\right)^{-1}X_{ml} \tag{808}$$
$$\hat{Y}_n = \mathbf{H}_{nm}Y_m \tag{809}$$

which, as you can see, puts the "hat" on our initial response observations, y_m. This smoothing matrix depends on an inversion, and as mentioned before most solvers will fail if the data matrix has too many features and not enough data points. But the way around this is through Bayesian methods. We'll start by noting that the likelihood from last post was the Likelihood of the data, given the model:

$$P(X,Y|\beta) \tag{810}$$

But, what if we'd like to write – for some, more intuitively and accurately – the likelihood of the model, given the data? This can be written as:

$$P(\beta|X,Y) = \frac{P(X,Y|\beta)P(\beta)}{P(X,Y)} \tag{811}$$

The second term on the numerator is something called a prior, and encodes our a priori beliefs on the values of β in our model. If we specify a Normal Prior, with some variance s, we get:

$$P(\beta) = \frac{1}{\sqrt{2\pi s^2}}e^{-\beta^2/2s^2} \tag{812}$$

The term $P(\beta|X,Y)$ is called the posterior, and represents our "new" beliefs on the model after accounting for the data that we have seen. Now, taking the log of the Posterior instead of the log of the likelihood – and ignoring the term in the denominator since it contains no dependence on β, we get:

$$\mathcal{L}(\beta|X,Y) = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{(\beta_d X_{nd} - Y_n)^2}{2\sigma^2} - \frac{1}{2}\log(2\pi s^2) - \frac{\beta_d\beta_d}{2s^2} + \mathcal{O}(X,Y) \tag{813}$$

Taking the gradient with respect to β now, we get an extra term in our equations:

$$\frac{\partial\mathcal{L}(\beta|X,Y)}{\partial\beta_d} = \frac{(\beta_l X_{nl} - Y_n)}{\sigma^2}X_{nd} + \frac{\beta_d}{s^2} \tag{814}$$
$$\frac{X_{nd}Y_n}{\sigma^2} = \beta_l\left(\frac{X_{nl}X_{nd}}{\sigma^2} + \frac{\delta^K_{ld}}{s^2}\right) \tag{815}$$
$$X_{nd}Y_n = \beta_l\left(X_{nl}X_{nd} + \delta^K_{ld}\frac{\sigma^2}{s^2}\right) \tag{816}$$
$$\beta_l = \left(X_{nl}X_{nd} + (\sigma/s)^2\right)^{-1}X_{nd}Y_n \tag{817}$$

We see that we've just got an extra term in the inverted matrix – namely the ratio of sample variance to prior variance – which adds to the diagonal of the feature l,d covariance estimate. What this does, practically speaking, is make the inverted matrix much more likely to be non-singular, and therefore resilient to have more features that datapoints, $D > N$.

As s get's smaller, what we essentially do is put isotropic, downward pressure on the β coefficients, pushing them down towards zero. This $L2$ norm or regularization on our model has lots of nice properties, and depending upon the strength of our prior, we can use it to protect against very "ill-defined problems", where $D >> N$.

The standard name for the method is called ridge regression, and people continue to be unaware of its benefits, such as protecting against co-linearity, and getting a sense of what regression coefficients do over varying strengths of regularization – called a regularization "path".

27.3 Part 3

As you can imagine, with increasing strength of the prior s, mentioned above, comes reduction in the magnitude of the regression coefficients β. Instead of using an $L2$ norm in the prior, one can also use an L1 norm, and then the log Likelihood becomes:

$$-\mathcal{L}(\beta|X,Y) = \frac{(X_{ni}\beta_i - Y_n)^2}{2\sigma^2} + \left(\sum_i |\beta_i|\right) \tag{818}$$

Which can be solved as a Quadratic programming problem as long as one puts an inequality constraint on the sum of the absolute coefficients of β: $T(\beta) = \sum_i |\beta_i| < t$. This type of normalization of the regression coefficients is called the LASSO, and by the nature of its β penalization, chooses solutions that are sparse in regressors – i.e. kills off coefficients and features that seem not to matter. With a decreasing value of t comes, fewer and few features, as you can imagine, and just like our parameter s above, we have explicit control over the "filtering" pressure of our regression coefficients β.

L2 and L1 regularization of regression coefficients lead to slightly different solutions. L2 tends to spread coefficient magnitude across clusters of variables that are all correlated with the target, while L1 aggressively prunes coefficient magnitude to the "winners" of the feature set. Making a plot of $\beta_i(t), \beta_i(s)$ for both L1 and L2 regularization, reveals this.

The lasso can be very useful when trying to isolate "what matters" in a regression problem, and just like ridge regression, helps control linear dependence and colinearity within the data matrix, but one can also use simple clustering techniques to choose the "best" set of features. For example, take the normalized correlation matrix:

$$\tilde{X}_{ni} = \frac{X_{ni} - \mu_i}{\sigma_i} \tag{819}$$

$$\rho_{ij} = \mathrm{corr}(x_i, x_j) = \frac{1}{N}\tilde{X}_{ni}\tilde{X}_{nj} \tag{820}$$

The upper diagonal portion of ρ_{ij} represents a graph, where the nodes are the features i and the edges are the matrix entries – each between 0 and 1. We can "cluster" our set of features very easily, by simply thresholding $\rho_{ij} > \epsilon$ and then picking out connected components from the matrix. The connected components – depending upon how many of them there are, each represent "feature groups", from which one can choose the most highly correlated feature with the target:

$$\{C_n\} = \mathrm{conn}(\rho_{ij} > \epsilon) \tag{821}$$

$$x_c = \mathrm{max}_{i\in c}\left(\mathrm{corr}(x_i, y)\right) \;\; \forall c \in \{C_n\} \tag{822}$$

Obviously, this filtered set of features – and their multiplicity – will be a function of ϵ. As $\epsilon \to 1$ we will have all features come out, $x_c = x_i$, and as $\epsilon \to 0$ we will have the single, most highly-correlated feature: $x_c = \max_i \mathrm{corr}(x_i, y)$.

The whole point of doing this, of course, is to find a set of features x_c that are statistically de-coupled from one - another, and it really reduces to a supervised down-sampling of the initial data.

Becaus regularization paths are so popular, especially from a diagnostic point of view it's worth mentioning that one of my favorite algorithms for sequentially adding in features to a regression problem is LARS – or least Angle Regression. The basic idea come from boosting, but I'll get to it in the next post.

27.4 Pricing Optimization

After reading through quite a bit of literature – or at least a weekend's worth – on optimized pricing and it seems as though the same ideas are being circulated, again and again and again. I know there are good resources out there in terms of pricing in high-volume environments, such as online advertising, but for the most part, in retail and macro "bidding", such as winning large contracts every few years or applying for an RFP, the thought process has remained the same: what's the probability of "winning" – i.e. getting the bid – at price point x, and what's the price at which we optimize the expected return. This can basically be described as:

$$P(y|\vec{x}) = \frac{1}{1+e^{-\beta\cdot\vec{x}}} \quad (823)$$

$$\mathrm{E}\,(x) = xP(y|x) \quad (824)$$

Where, I"ve already modeled the "winning" probability as a logistic regression – standard practice based on former papers. But, it's interested to note that supply and demand curves have a very close connection here, and most often this function $p(y|x)$ needs to have some specific properties, such as:

1. Be monotically decreasing in x – for non status-associated or "Giffen" goods.
2. Approach zero as $x \to \infty$.
3. Approach the total supply, call it D, as $x \to 0$.

A nice way to formulate this is of course with a right-sided, CDF. Integrating $p(y|x)$, what some people call a "willingness" to pay function:

$$d(x) = D\int_x^{\infty} dx\;\; p(y|x) \quad (825)$$

$$d(x) = Dp(y|X \geq x) \quad (826)$$

So the "demand" at price point x will now have some nice properties – such as being monotonically decreasing. When someone associates an "elasticity" with a supply and demand curve, such as:

$$d(x) \approx \alpha + \beta x \quad (827)$$

With $\beta < 0$, what you're actually doing is imposing a constant "willingness" to pay function, which is interesting because my "risk" of saying no to any deal – much like any consumer – is certainly not constant over all price points.

Typical strategies for pricing a single customer i've read have :

- Fit a logistic function / regression to the right-sided CDF, $p(y|X \geq x)$.
- Fit a linear regression to the demand function, $d(x)$

and then, using these "risk" curves, pointed out an optimal price x^*.

I've read have fit some form of the normalized demand curve, such

Which

28 Regression and Matching

Part IV
Useful Mathematical Functions

29 The Gamma Function Recursion Relation

29.1 For Integers

The factorial has the following property:

$$(z+1)! = (z+1)z! \tag{828}$$

Back in the 18th – maybe 17th – centuries, an interesting problem was posed: how do we represent the factorial of extremely large numbers, and, how do we compute it quickly?

Hence the gamma function, $\Gamma(z)$ which has the following definition and properties:

$$\Gamma(z) \equiv \int_0^\infty e^{-s}s^{z-1}ds \tag{829}$$

Integrating this function by parts, let us examine $\Gamma(z+1)$:

$$\begin{aligned}\Gamma(z+1) &= \int_0^\infty e^{-s}s^z ds &(830)\\ &= e^{-s}s^z|_0^\infty - z\int_{-\infty}^\infty e^{-s}s^{z-1}ds &(831)\\ &= z\Gamma(z) &(832)\end{aligned}$$

Interesting . . . if we examine $\Gamma(1)$, we find

$$\begin{aligned}\Gamma(1) &= \int_0^\infty e^{-s}ds &(833)\\ &= -e^{-s}|_0^\infty &(834)\\ &= 1, &(835)\end{aligned}$$

which means

$$\begin{aligned}\Gamma(n) &= 1\cdot 2\cdot 3\ldots(n-1) &(836)\\ &= (n-1)! &(837)\end{aligned}$$

Interesting. The gamma function of integer values yields the factorial of that integer value minus one.

29.2 For Half-Integers

Now what if we examine the Gamma function when n is a half-integer? Does the factorial of a half integer – or any non-integer for that matter – make sense? Let's take a look. Redefining s as x^2, we find

$$\Gamma(z) = \int_0^\infty e^{-s} s^{z-1} ds \tag{838}$$
$$= \int_0^\infty e^{-x^2} x^{2z-2} 2x dx \tag{839}$$
$$= 2\int_0^\infty e^{-x^2} x^{2z-1} dx \tag{840}$$

Setting $z = 1/2$,

$$\Gamma(\frac{1}{2}) = 2\int_0^\infty e^{-x^2} dx \tag{841}$$
$$= \int_{-\infty}^\infty e^{-x^2} dx \tag{842}$$
$$= \sqrt{\pi} \tag{843}$$

Citing our result from before, the integral above was simply the normalization of a Gaussian with variance $\sigma^2 = 1$. Or, $\langle x^0 \rangle$.

29.2.1 Relation of the Gamma Function to the moments of a Normal Distribution

Because the Normal distribution is even, it is a simple intuitive fact that the odd moments are zero. After all, an odd function times an even function is an odd function, and *that*, integrated over a symmetric domain will yield zero. But let us look at the even moments of a Gaussian:

$$\Gamma(z) = \int_0^\infty e^{-s} s^{z-1} ds \tag{844}$$
$$\Gamma(z) = 2\int_0^\infty e^{-x^2} x^{2z-1} dx \tag{845}$$
$$\Gamma(z+1/2) = 2\int_0^\infty e^{-x^2} x^{2z} dx \tag{846}$$
$$\Gamma(z+1/2) = \int_{-\infty}^\infty e^{-x^2} x^{2z} dx \tag{847}$$

Letting $x' = \frac{x'}{\sqrt{2}\sigma}$, we find $\frac{dx'}{\sqrt{2}\sigma} = dx$; and $(\frac{1}{2\sigma^2})^z (x')^{2z} = x^{2z}$:

$$\Gamma(z+1/2) = \int_{-\infty}^\infty e^{-(x')^2/2\sigma^2} (\frac{1}{2\sigma^2})^z (x')^{2z} \frac{dx'}{\sqrt{2}\sigma} \tag{848}$$
$$\sqrt{2}\sigma(2\sigma^2)^z \Gamma(z+1/2) = \int_{-\infty}^\infty e^{-(x')^2/2\sigma^2} (x')^{2z} dx' \tag{849}$$
$$\frac{\sqrt{2}(2\sigma^2)^z \Gamma(z+1/2)}{\sqrt{2\pi}} = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^\infty e^{-(x')^2/2\sigma^2} (x')^{2z} dx' \tag{850}$$
$$\frac{(2\sigma^2)^z \Gamma(z+1/2)}{\sqrt{\pi}} = \langle x^{2z} \rangle \tag{851}$$
$$\frac{2^n \Gamma(n+1/2)}{\Gamma(1/2)} (m_2)^n = m_{2n} \tag{852}$$

Pretty cool right? This means that all the moments of a Gaussian distribution – and thereby the entire taylor expansion in terms of its moments – can be created from the second moment. If one knows the variance, one knows everything . . . .

29.3 The Beta Function

29.4 The N-dimensional Ball

One of the coolest applications of the Gamma and Beta functions lie in the recursion relations for calculating solid angle and volume in arbitary dimensions. Let us begin by describing the volume of an n-dimensional sphere

$$V_n = \int r^{n-1} dr \int d\Omega_n \tag{853}$$

Note that I have split up the integral into it's radial portion – which goes from zero to infinity – and the solid angle portion – or, the surface area per unit radius (to the nth power minus one). This indexing with respect to n allows us to write a recursion relation, starting with the base case $n = 2$:

$$\begin{aligned} V_2 &= \int r dr \int d\Omega_2 && (854) \\ \int d\Omega_2 &= 2\pi && (855) \\ V_2 &= \frac{r^2}{2} 2\pi && (856) \\ V_2 &= \pi r^2 && (857) \end{aligned}$$

As expected. Now let us rewrite in polar coordinates, and create the n dimensional case in terms of the $n-1$ dimensional case:

$$V_n = \int r^{n-1} dr \int \sin^{n-2}(\theta) d\theta \int d\Omega_{n-1} \tag{858}$$

By taking successive steps down, we can eventually arrive at the case $n = 2$.

$$\begin{aligned} \int d\Omega_n &= \int \sin^{n-2}(\theta_{(n-1)}) d\theta_{(n-1)} \int \sin^{n-3}(\theta_{(n-2)}) d\theta_{(n-2)} \int d\Omega_{n-2} && (859) \\ &= \int \sin^{n-2}(\theta_{(n-1)}) d\theta_{(n-1)} \int \sin^{n-3}(\theta_{(n-2)}) d\theta_{(n-2)} \cdots \int d\Omega_2 && (860) \end{aligned}$$

Notice how I have indexed each theta coordinate with a subscript $n-1$, $n-2$, etc. We will call the radius the n^{th} coordinate. If we assume all of these integrands are separable – which they are if our function is isotropic, or spherically symmetric in n-dimensions – then we can write each of their limits explicitly:

$$\int d\Omega_n = \int_0^{\pi} \sin^{n-2}(\theta_{(n-1)}) d\theta_{(n-1)} \int_0^{\pi} \sin^{n-3}(\theta_{(n-2)}) d\theta_{(n-2)} \cdots \int_0^{2\pi} d\phi \tag{861}$$

By symmetry, we can cut these integration limits from zero to π in half, and equate each integrand to a beta function:

$$
\begin{aligned}
\int d\Omega_n &= \int_0^{\pi/2} 2\sin^{n-2}(\theta_{(n-1)})d\theta_{(n-1)} \int_0^{\pi/2} 2\sin^{n-3}(\theta_{(n-2)})d\theta_{(n-2)} \cdots \\
&\quad \int_0^{\pi/2} 2\sin(\theta_{(2)})d\theta_{(2)}(2\pi) \qquad (862) \\
\int d\Omega_n &= B(\frac{1}{2},\frac{n-1}{2})B(\frac{1}{2},\frac{n-1}{2})\cdots B(\frac{1}{2},1)2\pi \qquad (863) \\
\int d\Omega_n &= \Pi_{i=3}^{n} B(\frac{i}{2},\frac{i-1}{2})2\pi \qquad (864)
\end{aligned}
$$

The right hand side of this equation can be simplified into telescoping gamma functions, since

$$
B(x,y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}, \tag{865}
$$

as we saw from last section. Putting this into our expression for the n-dimensional solid angle,

$$
\begin{aligned}
\int d\Omega_n &= \Pi_{i=3}^{n} B(\frac{i}{2},\frac{i-1}{2})2\pi \qquad (866) \\
&= (\frac{\Gamma(\frac{1}{2})\Gamma(\frac{n-1}{2})}{\Gamma(\frac{n}{2}))}(\frac{\Gamma(\frac{1}{2})\Gamma(\frac{n-2}{2})}{\Gamma(\frac{n-1}{2})})\cdots(\frac{\Gamma(\frac{1}{2})\Gamma(\frac{4-1}{2})}{\Gamma(\frac{4}{2})})\frac{\Gamma(\frac{1}{2})\Gamma(\frac{3-1}{2})}{\Gamma(\frac{3}{2})}2\pi \qquad (867) \\
&= \frac{\Gamma(\frac{1}{2})^{n-2}}{\Gamma(\frac{n}{2})}2\pi \qquad (868) \\
&= \frac{2\pi\sqrt{\pi}^{n-2}}{\Gamma(\frac{n}{2})} \qquad (869) \\
\int d\Omega_n &= \frac{2\pi^{n/2}}{\Gamma(\frac{n}{2})} \qquad (870)
\end{aligned}
$$

Whew!. Now we can related this expression to the volume of an n-dimensional ball:

$$
\begin{aligned}
V_n &= \int r^{n-1}dr \int d\Omega_n \qquad (871) \\
&= \int r^{n-1}dr \frac{2\pi^{n/2}}{\Gamma(\frac{n}{2})} \qquad (872) \\
&= \frac{r^n}{n}\frac{2\pi^{n/2}}{\Gamma(\frac{n}{2})} \qquad (873) \\
&= \frac{2\pi^{n/2}}{\Gamma(\frac{n}{2}+1)}r^n \qquad (874)
\end{aligned}
$$

There might be a pesky factor of two running around in there, but for the time being, we'll keep it there.

29.5 Dimensional Regularization

For an isotropic integrand in $\mathbb{R}^n$, we can reduce its evaluation only to radial coordinates:

$$\begin{aligned} \int f(x) d^n x &= \int f(r) r^{n-1} dr \int d\Omega_n \\ &= \int f(r) r^{n-1} \frac{2\pi^{n/2}}{\Gamma(\frac{n}{2})} dr \end{aligned}$$

Part V
Stochastic Processes

30 Time Series Data

30.1 Linear Wold Representation and Green's Functions

Recently, I've been reading a great deal about time series data, and the various ways to represent it, model it, even forecast it. Seems a fundamental concept in time series data is the Wold decomposition theorem, which states that any time series $\{y_t\}$ can be represented by a linear process, or infinite order moving average representation:

$$y_t = \mu + \sum_{k=0}^{\infty} \psi_k \epsilon_{t-k} \quad (875)$$

$$\psi_0 = 1 \quad (876)$$

$$\sum_{k=0}^{\infty} |\psi_k|^2 < \infty \quad (877)$$

$$\epsilon_t \sim WN(0, \sigma^2) \quad (878)$$

What this basically says is that any process can be viewed as an integrated "reaction" to some exciting noise/random force ϵ_t. That underlying noise ϵ may or may not be constant over time, or have the same statistical properties over time, but we'll restrict ourselves to the simplest case where the noise is Gaussian.

This framing of a discrete time-series looks unbelievably similar to a Green's function, where the impulse response of some observable $y(t)$ is given by the solution to a linear differential equation. Let's take the langevin equation as an example:

$$m \frac{\partial v(t)}{\partial t} = -\gamma v(t) + \epsilon(t) \quad (879)$$

This models the velocity of a particle, $v(t)$ in a viscous medium. We can frame this equation of motion as a linear operator:

$$\left(1 + \frac{\gamma}{m} \frac{\partial}{\partial t}\right) v(t) = \frac{1}{m} \epsilon(t) \quad (880)$$

$$\hat{L} v(t) = \frac{1}{m} \epsilon(t) \quad (881)$$

If we take the Laplace Transform or Fourier transform of this equation, we can derive the inverse of our linear operator, $L^{-1}(t)$, which is the Green's function. Just for practice, I'll go through the motions all over again:

$$\left(1 + \frac{\gamma}{m}\partial_t\right) G(t) \quad = \quad \delta^D(t) \tag{882}$$

Where δ^D is the dirac delta function. Taking the fourier transform we get:

$$(1 + i\omega\frac{\gamma}{m})G(\omega) \quad = \quad 1 \tag{883}$$

We now have an algebraic equation rather than a differential one, and inverting becomes easy. Trouble is, we have to back transform:

$$G(\omega) \quad = \quad \frac{1}{(1 + i\omega\gamma/m)} \tag{884}$$

$$G(t) \quad = \quad \int \frac{d\omega}{2\pi}\frac{e^{i\omega t}}{1 + i\omega\gamma/m} \tag{885}$$

This integral can be computed via the residue theorem, but has a couple of tricks in it, due to the sign of t. If we restrict ourselves to positive t, we get:

$$G(t) \quad = \quad e^{-(\gamma/m)t} \tag{886}$$

So now, convolving our Green's function with the force function, $\epsilon(t)$, we get:

$$v(t) \quad = \quad \int_0^t dt' e^{-(t-t')/\tau}\epsilon(t') \tag{887}$$

31 The Watson-Nadaraya Estimator

32 Differential Regularizers

Often, when we are solving a regression problem, we are given the following functional:

$$J[f] \quad = \quad l\left[f, y\right] + \Omega\left[f\right] \tag{888}$$

Where l is some loss functional and Ω is some regularizer. Given some finite dataset, this might look like, in the case of squared loss:

$$X \quad = \quad \{\mathbf{x}_n\}_{n=1}^N \tag{889}$$

$$J \quad = \quad \sum_{n=1}^N \left(f(\mathbf{x}_n) - y_n\right)^2 + \Omega(f) \tag{890}$$

For a linear basis expansion model, we have the following:

$$f(\mathbf{x}) \quad = \quad \mathbf{w} \cdot \phi(x) \tag{891}$$

$$J(\mathbf{w}) \quad = \quad \sum_{n=1}^N \left(\mathbf{w} \cdot \phi(x_n) - y_n\right)^2 + \lambda|\mathbf{w}|^2 \tag{892}$$

where $\lambda|\mathbf{w}|^2$ plays the role of a prior over functions. The cost function in this example proportional to the negative log prior, want we have essentially:

$$-\log P(\mathbf{w}|X, \mathbf{y}, \phi) \quad \sim \quad J(\mathbf{w}) \tag{893}$$

Minimizing the cost with respect to $\mathbf{w}$ is same thing as finding the mode of the posterior, or Maximum a Posteriori. (MAP). We've already talked about how, for such a regularized regression problem, we can right the solution as a linear combination of kernels, centered at the data:

$$f(\mathbf{x}) \;=\; \sum_{n=1}^{N} \alpha_n K(\mathbf{x}, \mathbf{x}_n) \tag{894}$$

a manifestation of the representer theorem. But one important question to ask, given some regularization functional, is, what's the "best" Kernel? Let's take for example the regularizer:

$$J \;=\; \int dx\, (f(x) - y)^2 + \int dx |\frac{\partial^2 f(x)}{\partial x^2}|^2 \tag{895}$$

This Ω functional penalizes curvature in our fitting function $f(x)$, and we can note that what such a regularizer really is, is a prior over functions, since:

$$\begin{aligned} P(f(x^*)|f(X), \mathbf{y}) &= \frac{P(X, y|f)P(f)}{P(X, \mathbf{y})} && (896) \\ &= \frac{\exp\left[-\int dx\, (f(x) - y)^2 - \int dx |\frac{\partial^2 f(x)}{\partial x^2}|^2\right]}{P(f(X), \mathbf{y})} && (897) \\ & && (898) \end{aligned}$$

We see that the prior on function is:

$$P[f] \;\sim\; \exp\left(-\int |f''(x)|^2 dx\right) \tag{899}$$

and, to be more general, we could have written the prior on our functions as a superposition of differential operators:

$$P[f] \;\sim\; \exp\left(-\int \sum_{m=1}^{\infty} a_m \frac{\partial^m}{\partial x^m} |f(x)|^2 dx\right) \tag{900}$$

If we integrate by parts, we note that this prior functional can be put into the form:

$$P[f] \;\sim\; \exp\left(-\int dx dx' f(x) K(x, x')^{-1} f(x')\right) \tag{901}$$

Which of course gives us the familiar prior assumptions:

$$\langle f(x)\rangle = 0 \tag{902}$$

$$\mathrm{Var}\left(f(x)\right) = K(x,x') \tag{903}$$

But, for a given differential regularizer, how do we find the associated Kernel? The answer is simple, it's just the Green's function of the operator:

$$\hat{L} = \sum_m a_m \frac{\partial^m}{\partial x^m} \tag{904}$$

$$\hat{L}K = \sum_m a_m \frac{\partial^m}{\partial x^m} K(x,x') = \delta(x-x') \tag{905}$$

An easy way to get the green's function – or in this case Kernel – is to fourier transform. We can re-write our prior in s space:

$$f(s) = \int dx e^{-isx} f(x) \tag{906}$$

$$P[f] \sim \exp\left(-\int ds \sum_{m=1}^{\infty} a_m |\mathbf{s}\cdot\mathbf{s}|^m |f(s)|^2 dx\right) \tag{907}$$

We see now the fourier transform of our inverse kernel is:

$$\frac{1}{K(s,s')} = \sum_m a_m (-1)^m |\mathbf{s}\cdot\mathbf{s}|^m \delta^D(\mathbf{s}+\mathbf{s}') \tag{908}$$

We see that the kernel is diagonal and in s space and semi-positive definite. Which means that we are translationally invariant in x space. We have:

$$K(x,x') = \int ds ds' e^{isx+is'x'} \frac{1}{\sum_m a_m (-1)^m |\mathbf{s}\cdot\mathbf{s}'|^m} \delta^D(\mathbf{s}+\mathbf{s}') \tag{909}$$

$$K(x-x') = \int ds e^{is(x-x')} \frac{1}{\sum_m a_m |\mathbf{s}\cdot\mathbf{s}|^m} \tag{910}$$

We see that, indeed, our Kernel will be translationally invariant, and when we put a prior over functions, what is essentially a LAGRANGIAN in physics:

$$\Omega[f] \sim L = \int dx \sum_m a_m \frac{\partial^m}{\partial x^m} f(x) \tag{911}$$

We find that the kernel is just the correlation function – or, the PROPAGATOR – for the resulting stochastic process. One example is the massive free particle in higher dimensions – or the free process in higher dimensional feature space – for which we get the Yukawa potential:

$$\Omega[f] \sim L = \int dx \frac{m^2}{2} f(x)^2 + \frac{1}{2}\nabla^2 f(x) \tag{912}$$

$$K(x-x') \sim \frac{1}{|x-x'|} e^{-|x-x'|m} \tag{913}$$

So, the Kernel we use when interpolating some field, or, specifying the prior of our Gaussian process, can be viewed as the Green's function to our penalizing "regularizer". If we want smooth functions up to order $m = M$, we know precisely how to formulate the associated $K(x - x')$. Note that all differential regularizers, such as discussed here will lead to stationary kernels and thus stationary processes.

Part VI
Cosmology

33 Assumptions

Let us imagine a universe with a density field $\rho(\vec{x})$ and mean density $\overline{\rho}$. Let us define the mass density perturbations from the mean as a

$$f(\vec{x}) = \frac{\rho(\vec{x}) - \overline{\rho}}{\overline{\rho}}. \tag{914}$$

This f is most often written as δ, but I do not want to confuse our density perturbations with the dirac delta function. Let us assume that the universe "chooses" to inhabit each point in space with a certain amount of overdensity, according to a Gaussian probability density function. This would imply – if such a Gaussian probability density function was centered – that the mean density is zero. Further, let us assume that the "choice" to inhabit one point is entirely statistically independent from another point's *choice*. Or at $t = 0$ – and by that we mean the real time equals zero – if such a thing exists – every single point in space performs an experiment, governed by a Gaussian probability density function, whose observable is the overdensity f.

To study the statistical properties and likelihoods of our three-dimensional density *field*, let us imagine that our universe has been "pixellated" into unit cubes of side length $s \sim \sqrt{\frac{\hbar G}{c^3}}$, of one Planck length. This is, in essence, the most absurd and refined N-Body simulation possible. Our density field $\rho(\vec{x})$ is now a single dimensional vector of N components. Or, building on our discussion from before, an ensemble of N simultaneous experiments at $t = 0$. This is a spatial ensemble of experiments, since we are comparing point to point within our universe.

33.1 Absurd Multinomial

One easy way to describe this pixellated universe would be with multivariate Gaussian – where, the number of variables is the number of pixels in the Universe, a ridiculous number, and we thus measure the plausibility of a random vector, or set of outcomes. Let those set of outcomes be labeled $f(x_i) \to x_i$, denoting the overdensity at a single point in space:

$$P(\mathbf{x}) = \frac{e^{-\mathbf{x}\cdot\mathbf{C}^{-1}\cdot\mathbf{x}}}{(2\pi)^{N/2}\sqrt{|\mathbf{C}|}} \tag{915}$$

Where, $\mathbf{C}$ is the covariance matrix; its components are given by the cross correlation estimators:

$$\begin{aligned} \mathbf{C}_{ij} &= \langle x_i x_j \rangle && (916) \\ &= \frac{\partial^2 \phi(\mathbf{k})}{\partial k_i \partial k_j}\Big|_{\mathbf{k}=\vec{0}} && (917) \end{aligned}$$

Where, $\phi(\mathbf{k})$ is the characteristic function – fourier transform – of our multinomial $P(\mathbf{x})$, and the angled brackets denote an ensemble average over parallel universes (more on that soon).

If we assume Gaussian initial conditions, then this probability density function above perfectly describes the likelihood of various over densities. And if we assume statistically independent values of x_i then the matrix $\mathbf{C}$ will be diagonal.

This will not always be true however, as universe evolves.

33.2 Ergodicity

Now, if we would like to measure the first moment, or the expectation value of density, we would have to examine a single point in space – let's call it f_i. But if we sit there, during the evolution of the universe and watch f_i, we'll find that its "choice" has been made; the single pixel has "chosen" to be overdense, and so now the characteristics of f_i are determined by Gravity. Perhaps we had better think of the prior probability density function of f_i before it made its choice.

But how do we get information about such a PDF? How could we measure the width or expectation value of a Gaussian that existed once but has now collapsed into a delta function at $t = 0$? One could imagine many parallel universes, where a scientist is *present* at the beginning of each one, ready to measure the same point's – f_i's – "choice" to be a certain density value. We would expect this ensemble of density measurements across parallel universes to have the properties of a Gaussian. Unfortunately, we only get one universe.

But we have another ensemble to rely upon, and that is the prenominal spatial ensemble of density values at $t = 0$. Is it possible that we can derive the same information from the spatial ensemble as we can from the parallel universe ensemble (both at $t = 0$)?

The answer lies in ergodicity. If one assumes that the spatial average, over all the separate points in our universe will elicit the same properties as the parallel universe *ensemble* average, then we say the field is ergodic. Going with our assumptions from above, we expect every single point's density "choice" in space to be statistically independent: for such a set up, the spatial characteristics will be Gaussian. So in essence, our spatial measurements could inform us as to the initial probability distribution, governing each point in space's "choice".

34 The Two Point Correlation Function

The correlation between two functions of a single variable is defined as

$$(f \star g)(y) \quad = \quad \int f(x-y)g(x)dx. \tag{918}$$

In many ways, this represents the overlap of signals, probability densities, or simple function characteristics, depending upon your interpretation. We are interested in finding the statistical correlation between two points in our universe being overdense. [Or, for complete generalization, we are interested in the correlation of two random functions of a parametrized variable $f(\vec{x})$, which is a tuple.]

Let us use the density perturbation $\delta(\vec{x}) = \frac{\delta(\vec{x}-\bar{\delta}}{\bar{\delta}}$ as our random function of interest. Let's examine the spatial – not the ensemble – average of two separate points, decomposed into Fourier components:

$$\begin{aligned}
\langle \delta(\vec{x})\delta(\vec{y})\rangle \quad &= \quad \langle \int\int \delta(\vec{k})\delta^\star(\vec{k'})e^{i\vec{k}\cdot\vec{x}}e^{-i\vec{k'}\cdot\vec{y}}d^3kd^3k'\rangle && (919)\\
&= \quad \int\int \langle\delta(\vec{k})\delta^\star(\vec{k'})\rangle e^{i\vec{k}\cdot\vec{x}}e^{-i\vec{k'}\cdot\vec{y}}d^3kd^3k' && (920)\\
& && (921)
\end{aligned}$$

Where in the second line, I have taken the expectation value brackets inside the integral because the exponential factors are not stochastic.

If, under translations, this ensemble average is invariant, we can say that the the correlation is only a function of their relative separation:

$$\begin{aligned} \langle\delta(\vec{x})\delta(\vec{y})\rangle &= \langle\delta(\vec{x}+\vec{a})\delta(\vec{y}+\vec{a})\rangle,\ \forall\vec{a} && (922)\\ \Rightarrow \langle\delta(\vec{x})\delta(\vec{y})\rangle &= \int\int \delta(\vec{x})\delta(\vec{x}-\vec{y})d^3xd^3y && (923)\\ \xi(\vec{y}) &= \int \delta(\vec{x})\delta(\vec{x}-\vec{y})d^3x && (924)\\ \langle\delta(\vec{x})\delta(\vec{y})\rangle &= \langle\xi(\vec{x}-\vec{y})\rangle && (925)\\ \langle(\delta\star\delta)(\vec{r})\rangle &= \langle\xi(\vec{r})\rangle && (926) \end{aligned}$$

Where I am now using the more common representation of the two point correlation function, $\xi(\vec{r})$, where $\vec{r}=\vec{x}-\vec{y}$. We can now define the autocorrelation between density perturbations:

$$(\delta\star\delta)(\vec{r}) = \xi(\vec{r}) \qquad (927)$$

Let us examine this property of translational invariance – or homogeneity – in frequency space:

$$\begin{aligned} \langle\delta(\vec{x})\delta(\vec{y})\rangle &= \langle\delta(\vec{x}+\vec{a})\delta(\vec{y}+\vec{a})\rangle,\ \forall\vec{a} && (928)\\ \overline{\langle\xi(\vec{y})\rangle} &= (\delta\star\delta)(\vec{y}) && (929)\\ \overline{\langle\xi(\vec{y})\rangle} &= \int\int\langle\delta(\vec{k})\delta^\star(\vec{k'})\rangle e^{i\vec{k}\cdot\vec{x}}e^{-i\vec{k'}\cdot\vec{y}}e^{i\vec{a}\cdot\vec{k}-\vec{k}'}d^3kd^3k',\ \forall\vec{a} && (930)\\ \overline{\langle\xi(\vec{y})\rangle} &= \langle\delta(\vec{k})\delta(\vec{k'})\rangle\delta^D(\vec{k}-\vec{k}') && (931) \end{aligned}$$

Where δ^D is the dirac delta function – as compared to the density perturbation field δ. This final equation shows that separate Fourier modes are uncorrelated in a homogeneous universe. We can further refine this equation by stating the two point correlation function is invariant under rotations, thereby asserting an isotropic density field. We find that the two point correlation function in both frequency and x space is a function of a single, scalar variable. Let us rewrite our equations from before:

$$\begin{aligned} \xi(|\vec{x}-\vec{y}|) &= (\delta\star\delta)(|\vec{x}-\vec{y}|) && (932)\\ &= \int \delta(\vec{k})\delta^*(\vec{k})e^{i\vec{k}\cdot(\vec{x}-\vec{y})}d^3k && (933)\\ &= \int |\delta(\vec{k})|^2e^{i\vec{k}\cdot(|\vec{x}-\vec{y}|)}d^3k && (934)\\ &= \int |\delta(\vec{k})|^2e^{i\vec{k}\cdot(\vec{x}-\vec{y})}k^2\sin\theta d\theta dkd\phi && (935) \end{aligned}$$

If we set $\vec{x}-\vec{y}$ along the k_z axis, we can simplify this integral, writing the dot product $\vec{k}\cdot(\vec{x}-\vec{y}) = kr\cos\theta$, we find

$$\begin{aligned} \xi(r) &= \int |\delta(\vec{k})|^2e^{ikr\cos\theta}k^2\sin\theta d\theta dkd\phi && (936)\\ &= 2\pi\int |\delta(\vec{k})|^2e^{ikr\cos\theta}k^2d(\cos\theta)dk && (937)\\ &= 2\pi\int_0^\infty |\delta(\vec{k})|^2(\frac{\sin(kr)}{kr})k^2dk && (938)\\ \xi(r) &= 2\pi\int_0^\infty P(k)(\frac{\sin(kr)}{kr})k^2dk && (939) \end{aligned}$$

We refer to $P(k)$ or $|\delta(k)|^2$ as the spectral density – or the Power spectrum – whose integral over all of k space is equal to the two point correlation function ξ's integral over all of x-space by the correlation theorem. This spectral decomposition is quite common in Gaussian Random Field theory, and is simply a special case for three-dimensional, isotropic and homogeneous fields.

Notice that $\xi(0)$ is equal to the variance of the spatial density distribution – assuming zero mean. (Missing factor of two here!) Thus we find:

$$\begin{aligned}
\lim_{r\to 0}\xi(r) &= 2\pi\int_0^\infty P(k)k^2dk && (940)\\
\sigma^2 &= \int_0^\infty 2\pi P(k)k^2dk && (941)\\
&= \int_0^\infty (2\pi P(k)k^3)d\ln k && (942)\\
\sigma^2 &= \int_0^\infty \Phi(k)d\ln k && (943)
\end{aligned}$$

Where is $\Phi(k) = 2\pi k^3 P(k)$ is the dimensionless quantity, representing the total amplitude ... (fill in later)

35 Power Spectrum Estimators

An important thing to note, the square of a Gaussian random variable follows a Rayleigh distribution. Peacock and Nicholson (1991) write the fourier coefficients of the overdensity field as:

$$a_k = \sum_i \frac{n_i}{N}e^{i\mathbf{k}\cdot\mathbf{x}_i} \qquad (944)$$

This is a sum of random variables, and so in the limit $N\to\infty$, a_k is drawn from a Gaussian. Meaning, if we have no correlation between different cell ($i\neq j$) occupation numbers:

$$\begin{aligned}
a_k a_k' &= \sum_{i,j}\frac{n_in_j}{N^2}e^{i\mathbf{k}\cdot\mathbf{x}_i+i\mathbf{k}'\cdot\mathbf{x}_j} && (945)\\
\langle a_k a_k'\rangle &= \sum_{i,j}\frac{\delta_{ij}}{N^2}e^{i\mathbf{k}\cdot\mathbf{x}_i+i\mathbf{k}'\cdot\mathbf{x}_j} && (946)\\
\langle a_k a_k'\rangle &= \sum_{i}\frac{1}{N^2}e^{i(\mathbf{k}+\mathbf{k}')\cdot\mathbf{x}_i} && (947)\\
\langle a_k a_k'\rangle &= \frac{1}{N}\delta_{k,-k'} && (948)\\
\langle |a_k|^2\rangle &= \frac{1}{N} && (949)
\end{aligned}$$

So, we have the square of a random variable, who is drawn from some Gaussian distribution. Thus $|a_k|^2$ will be drawn, in the absence of clustering/gravity, from a Rayleigh distribution:

$$P(\langle|a_k|^2\rangle > X) = e^{-NX} \qquad (950)$$

This is the 'Poisson noise' that will undermine – or as Peacock and Nicholson say, 'overlay' – the clustering signal that we are trying to measure. In the case of non-zero correlation we have to split things into two separate sums:

$$\langle a_k a_k' \rangle = \frac{1}{N}\delta_{k,-k'} + \frac{1}{N^2}\sum_{i\neq j} C_{ij} e^{i\mathbf{k}\cdot\mathbf{x}_i + i\mathbf{k}'\cdot\mathbf{x}_j} \tag{951}$$

Somehow, we need to incorporate translational invariance into the second term:

$$\begin{aligned} \langle |a_k|^2 \rangle &= \frac{1}{N} + \frac{1}{N^2}\sum_{i\neq j} C_{ij} e^{i\mathbf{k}\cdot(\mathbf{x}_i - \mathbf{x}_j)} && (952) \\ &= \frac{1}{N} + \frac{N-1}{N}\langle |\delta_k^{\text{cluster}}|^2 \rangle && (953) \end{aligned}$$

Where the last line is a little spurious, but I"m just following Peacock & Nicholson (1991) to the T. So the power spectrum estimator must subtract off this shot noise:

$$P(k) \approx \langle |a_k|^2 \rangle - \frac{1}{N} \tag{954}$$

Note that the amount of power in a single "bin" k – or spherical shell in k-space – is equal to the integral:

$$\begin{aligned} &\frac{V}{(2\pi)^3}\int_k |\delta(q)|^2 d^3q && (955) \\ &\frac{V}{(2\pi)^3}\int_k P(q) d^3q && (956) \end{aligned}$$

As the survey size V increases, we expect the amount of power in this bin to remain the same – why? – and so find that the power spectrum scales with inverse volume. (There are probably better ways to argue this, just by dimensional analysis.) Thus, our signal to noise ratio, for shot noise goes like $N/V = n$. If we are to create estimators on the overdensity field, given our discrete data sample, we may use – and perhaps, should use – reciprocal variance weighting, which means:

$$\hat{\delta}(\mathbf{x}) = \frac{\delta(\mathbf{x})n_b(\mathbf{x})}{\langle n_b \rangle} = \delta(\mathbf{x})W(\mathbf{x}) \tag{957}$$

Where I've written background density as $n_b(\mathbf{x}) = \langle n \rangle W(\mathbf{x})$. For any expectation values with respect to $\delta(\mathbf{x})$, we should use $\delta(\mathbf{x})W(\mathbf{x})$. The fourier transform becomes – which is now a weighted estimator of the fourier coefficient:

$$\begin{aligned} \hat{\delta}(\mathbf{k}) &= \frac{1}{V}\int d^3x W(\mathbf{x})\delta(\mathbf{x})e^{i\mathbf{k}\cdot\mathbf{x}} && (958) \\ &= (W * \delta)(\mathbf{k}) && (959) \end{aligned}$$

So in Fourier space – as expected – this multiplication becomes a convolution. This estimator on the Fourier coefficient, given some discrete sample over varying background density – non-uniform $W(\mathbf{x})$, non dirac delta $W(\mathbf{k})$ – might be biased, in the sense that:

$$|\hat{\delta}(\mathbf{0})|^2 = \frac{1}{V^2}\left(\int d^3x W(\mathbf{x})\delta(\mathbf{x})\right)^2 \neq 0 \tag{960}$$

So the Power spectrum at $k = 0$ – the DC component – is non-zero. This may be undesirable, and can be corrected by just subtracting off a constant:

$$\hat{\delta}(\mathbf{x}) \;=\; W(\mathbf{x})\left(\delta(\mathbf{x}) - \int d^3x W(\mathbf{x})\delta(\mathbf{x})\right) \tag{961}$$

36 Interpolation of the Bispectrum and decomposition into Multipoles

37 Eulerian Fluid Dynamics

Ok, so let's begin our massive review of Fluid dynamics in the Eulerian regime. We have the three equations:

$$a \tag{962}$$

37.1 Equations of Motion

37.2 Comoving Equations of Motion

We begin with the co-moving equations of motion for the overdensity field, $\delta(\mathbf{x})$.

$$\begin{aligned}\dot{\delta} + \frac{1}{a}\vec{\nabla}\cdot[(1+\delta)\mathbf{v}] &= 0 && (963)\\ \dot{\mathbf{v}} + \frac{1}{a}\left(\mathbf{v}\cdot\vec{\nabla}\right)\mathbf{v} + \frac{\dot{a}}{a}\mathbf{v} &= \frac{-\vec{\nabla}p}{\rho a} - \frac{1}{a}\vec{\nabla}\phi && (964)\\ \Delta\phi &= 4\pi G\bar{\rho}a^2\delta && (965)\end{aligned}$$

Linearizing these equations, we can throw out all terms that are of higher order in $\mathbf{v}$ or δ. Let us call the divergence of the velocity field θ.

$$\begin{aligned}\dot{\delta} + \frac{\theta}{a} &= 0 && (966)\\ \dot{\mathbf{v}} + \frac{\dot{a}}{a}\mathbf{v} &= \frac{-\vec{\nabla}p}{\rho a} - \frac{1}{a}\vec{\nabla}\phi && (967)\\ \Delta\phi &= 4\pi G\bar{\rho}a^2\delta && (968)\end{aligned}$$

If we take the divergence of the second equation and assume zero pressure, we can write:

$$\ddot{\delta} + 2H\dot{\delta} \;=\; 4\pi G\bar{\rho}\delta \tag{969}$$

This is the linear overdensity equation, which can be solved given a specific $a(t), a(z), H(z)$.

37.2.1 Worked examples

37.2.2 Hubble as Decaying Mode

Taking a look at the Friedmann equations:

$$\left(\frac{\dot{a}}{a}\right)^2 = H^2 = \frac{8\pi G}{3}\rho + \frac{K}{a^2} + \Lambda/3 \tag{970}$$

$$\rho_c = \frac{3H_0^2}{8\pi G} \tag{971}$$

$$H^2 = H_0^2\left(\frac{\Omega_m}{a^3} + \frac{\Omega_K}{a^2} + \Omega_\Lambda\right) \tag{972}$$

Differentiating with respect to time, we find

$$2H\dot{H} = H_0^2\left(-3\frac{\Omega_m}{a^3} + -2\frac{\Omega_K}{a^2} + \Omega_\Lambda\right)H \tag{973}$$

$$\dot{H} = \frac{H_0^2}{2}\left(-3\frac{\Omega_m}{a^3} + -2\frac{\Omega_K}{a^2} + \Omega_\Lambda\right) \tag{974}$$

$$\ddot{H} = \frac{H_0^2}{2}\left(9\frac{\Omega_m}{a^3} + 4\frac{\Omega_K}{a^2} + \Omega_\Lambda\right)H \tag{975}$$

$$\tag{976}$$

Now examining $\ddot{H} + 2H\dot{H}$ we find

$$\ddot{H} + 2H\dot{H} = H_0^2 H\left(\frac{3}{2}\Omega_m a^{-3}\right) \tag{977}$$

This right hand side can viewed by examining the following

$$H_0^2 = \frac{8\pi G}{3}\rho_c \tag{978}$$

$$\frac{1}{\rho_c} = \frac{8\pi G}{3H_0^2} \tag{979}$$

$$\frac{3}{2}\frac{\rho}{\rho_c} = \frac{4\pi G}{H_0^2}\rho \tag{980}$$

$$\frac{3}{2}\Omega_m = \frac{4\pi G}{H_0^2}\rho \tag{981}$$

$$\frac{3}{2}\Omega_m H_0^2 = 4\pi G\rho \tag{982}$$

$$\tag{983}$$

And so we find that the Hubble constant satisfies the same differential equation as our overdensity field:

$$\ddot{H} + 2H\dot{H} = 4\pi G\rho H \tag{984}$$

$$\ddot{\delta} + 2H\dot{\delta} = 4\pi G\bar{\rho}\delta \tag{985}$$

We have studied the specific case of a universe dominated by Cold Dark Matter and Dark Energy. But what is interesting to note is that in such a case, the Hubble constant always corresponds to the decaying mode, $D_-(z)$.

37.2.3 Jean's Length

Now let there be a non-zero pressure in the our overdensity model.

38 Lagrangian Fluid Dynamics

In order to formulate fluid mechanics from a Lagrangian, or phase space perspective, we begin with the ansatz, or maybe the one-to-one mapping

$$\mathbf{x} = \mathbf{q} + \psi \tag{986}$$

Now, the hope is that this mapping is always invertible, i.e.

$$\rho d^3x = \rho_0 d^3q \tag{987}$$
$$\rho ||\mathbf{J}_{ij}|| d^3q = \rho_0 d^3q \tag{988}$$
$$||\mathbf{J}_{ij}||^{-1} = 1 + \delta(\mathbf{x}, t) \tag{989}$$
$$\det\left(\frac{\partial \psi_i}{\partial q_j}\right)^{-1} = 1 + \delta(\mathbf{x}, t) \tag{990}$$

Now this determinant might be singular, and in such a case we expect an infinite density at some $\mathbf{x}$. This is called stream crossing. It can be noted, in a potential flow, where, we can write

$$\mathbf{x} = \mathbf{q} + t\mathbf{v} \tag{991}$$

By noting that now we have:

$$\frac{1}{\det\left(\delta_{ij} + t\frac{\partial v_i}{\partial x_j}\right)} = 1 + \delta(\mathbf{x}, t) \tag{992}$$

We know that deformation matrix $\mathcal{D}_{ij} = \frac{\partial v_i}{\partial x_j}$ to be symmetric, since we can write $v_i \sim \partial_i \phi$, for some potential, and therefore it is diagonalizable. In such a case we have real eigenvalues and eigenvectors,

$$\frac{1}{(1 + t\lambda_1)(1 + t\lambda_2)(1 + t\lambda_3)} = 1 + \delta(\mathbf{x}, t) \tag{993}$$

Once one of these factors on the bottom hit zero we will have a huge amount of density. The eigenvalues λ_i are ordered above $\lambda_1 > \lambda_2 > \lambda_3$ and are very likely to be **simultaneously** of the same value. When one eigenvalue signifcantly leads the others, we have what's called pancake formation. When one significantly lags, we have string formation. This topological speech has been ramped up for many years by many people, in particular Prof. Shandarin. It is important to note from a topological point of view, those eigenvectors correspond to 3 principle radii of curvature in an equal-time slice in spacetime. [3]

[3] Note to self, I should look more into this

38.1 Grintein and Wise

Some of the cool stuff about the Zeldovich approximation can be seen if we write our lagrangian displacement vector as essentially a velocity with some growing mode placed next to it:

$$\mathbf{x} = \mathbf{q} + b(t)\psi(q) \tag{994}$$

Then we have, taking the derivative with respect to time and keeping $\mathbf{q}$ – which end up being our co-moving coordinates fixed:

$$\partial_t \mathbf{x}|_q = \mathbf{v}_p = \dot{b}\psi(q) \tag{995}$$

And so, if we want to write a phase distribution in the comoving coordinates, we can write:

$$f(\mathbf{q}, \mathbf{v_p}, t) = \frac{\langle\rho\rangle}{m}\delta^3\left(\mathbf{v}_p - \dot{b}\psi(q)\right) \tag{996}$$

39 Relation to The Gaussian Ensemble

This variance, if we assume ergodicity of the spatial density distribution and Gaussian random phase initial conditions, fully specifies the original normal distribution that each density value was picked from, since:

$$\sigma^2 = 2\pi\int_0^\infty P(k)k^2dk = \xi(0) \tag{997}$$

$$\frac{2^n\Gamma(n+1/2)}{\Gamma(1/2)}(\sigma^2)^n = m_{2n}. \tag{998}$$

This means that under these conditions, if one has the power spectrum, one can reconstruct the entire initial density distribution.

Part VII
Mathematical Cookbook

40 Sturm Liouville Problems

The following three problems are equivalent:

1. The sturm-liouville problem in differential equation form

$$-\frac{d}{dx}\left(p(x)y'\right) + q(x)y = \lambda w(x)y \tag{999}$$

   Where prime denotes differentiation with respect to x and $w(x), p(x)$ do not vanish on the interval of interest.

2. Extremizing the following functional $F[y]$

$$F[y] \quad = \quad \int_a^b \left(p(x)(y')^2 + q(x)y^2\right) dx \tag{1000}$$

   subject to the constraint

$$G[y] \quad = \quad \int_a^b wy^2 dx = 1 \tag{1001}$$

   The eigenvalues above are equal to the values of $F[y]$.

3. Finding the functions for which

$$\Lambda[y] \quad = \quad \frac{F[y]}{G[y]} \tag{1002}$$

   is stationary. The eigenvalues of our first problem are then given by the values of $\Lambda[y]$.

Our third option is called the Rayleigh quotient, and relies upon the boundary conditions of our function of interest, y, being fixed at the endpoints a, b.

40.1 Variational to Dif EQ

To show that these problems are equivalent, consider

$$\delta(F - \lambda G) \quad = \quad \int \left(p(x)(y')^2 + q(x)y^2 - \lambda wy^2\right) dx \tag{1003}$$

Our effective Lagrangian is

$$L \quad = \quad p\frac{dy}{dx}^2 + y^2(q - \lambda w) \tag{1004}$$

yielding an equation of motion

$$\frac{d}{dx}\left(2p\frac{dy}{dx}\right) \quad = \quad 2y(q - \lambda w) \tag{1005}$$

which simplifies – after cancelling the 2's – to our Sturm-Liouville Problem. Notice that the lagrange multiplier is our eigenvalue.

40.2 Dif EQ to Variational Method

Multiplying the Sturm-Liouville equation and integrating both sides, we find

$$\int -y\frac{d}{dx}\left(py'\right)+qy^2dx \quad = \quad \lambda\int wy^2dx \tag{1006}$$

$$\left(ypy'\right)|_a^b+\int\left(p(y')^2+qy^2\right)dx \quad = \quad \lambda G[y] \tag{1007}$$

$$F[y] \quad = \quad G[y]\lambda \tag{1008}$$

So, without the normalization condition $G[y]=1$, we find that the $\Lambda[y]=\lambda$. oherwise, our functional $F[y]$ yields the eigenvalues.

40.3 Sturm-Liouville Appropriate Boundary Conditions

If we represent the SR differential equation as an operator on y, we could possible write our extension above as

$$\mathcal{L}[y] \quad = \quad \lambda wy \tag{1009}$$

$$\int y\mathcal{L}[y]dx \quad = \quad \int\lambda wy^2dx \tag{1010}$$

$$\int y\mathcal{L}[y]dx \quad = \quad \lambda G[y] \tag{1011}$$

$$\frac{\int y\mathcal{L}[y]dx}{G[y]} \quad = \quad \lambda \tag{1012}$$

Now if we have two functions p,q that are real, and we have new solution ϕ, that is possibly complex, we can write $\mathcal{L}[\phi]^\star=\mathcal{L}[\phi^\star]$:

$$\mathcal{L}[\phi] \quad = \quad -\frac{d}{dx}\left(p\phi'\right)+q\phi=\lambda w(x)\phi \tag{1013}$$

Let us examine the following:

$$\int f\mathcal{L}[\phi]dx \quad = \quad -fp\phi'|_a^b+\int\left(p\phi'f'+qf\phi\right)dx=\lambda\int w(x)f\phi dx \tag{1014}$$

If f is another complex valued function, we can compare the following two integrals, to construct another Green's theorem (I feel like everything is called Green's theorem):

$$\int u^\star\mathcal{L}[v]dx-\int\mathcal{L}[\phi^\star]vdx \quad = \quad -\left(u^\star pv'-vp(u^\star)'\right)|_a^b \tag{1015}$$

If the right hand side of this equation is zero, or if we require

$$p(x)\left(u^\star\frac{dv}{dx}-v\frac{du^\star}{dx}\right)|_a^b \quad = \quad 0 \tag{1016}$$

then we can write

$$\int u^\star \mathcal{L}[v]dx = \int \mathcal{L}[\phi^\star]vdx \tag{1017}$$

$$\int \phi_1^\star \mathcal{L}[\phi_2]dx = \int \mathcal{L}[\phi_1^\star]\phi_2 dx \tag{1018}$$

$$\lambda_2 \int w(x)\phi_1^\star \phi_2 dx = \lambda_1^\star \int w(x)\phi_2^\star \phi_1 dx \tag{1019}$$

Which is true if and only if the two functions ϕ_1, ϕ_2 are orthogonal to each other with respect to the norm – or weight function – $w(x)$. Or if the eigenvalues satisfy $\lambda_2 = \lambda_1^\star$.

But wait! If we set $\phi_1 = \phi_2$, then we find that all eigenvalues must be real. And so two separate functions that satisfy our SR equation, given different eigenvalues, must be orthogonal.

40.3.1 Examples

Here are some examples and exercises that may be useful. The following differential equations can be written in SR form:

1. $\phi'' - 2x\phi' = -\lambda\phi$ (Hermite's Equation)
2. $(1 - x^2)\phi'' - x\phi' = -\lambda\phi$ for $-1 < x < 1$ (Chebyshev's Equation)
3. $x\phi''_(1 - x)\phi' = -\lambda\phi$ for $0 < x$ (Laguerre's Equation)
4. $\phi'' = -\lambda\phi$ for $0 < x < L$ (SHO)
5. $x^2\phi'' + x\phi' + (x^2 - \lambda^2)\phi = 0$ (Bessel's Equation) Which can be reduced to, if one divides by x

   $$[x\phi']' + x\phi = \frac{\lambda^2}{x}\phi \tag{1020}$$

   Interestingly, the weight function for Bessel functions is $1/x$.
6. $(1 - x^2)\phi'' - 2x\phi' = -\lambda(\lambda + 1)\phi$ (Legendre's equation). Which leads to $q(x) = 0$, $p(x) = 1 - x^2$ and $w(x) = 1$.

41 Green's Functions and Propagators

Say