Lecture 13

- Gaussian RVs continued
- Q function continued
- Expected Value
- Moments

Feedback Evaluation Form

Fill this out! Note: you need to be signed into Chrome or web browser through your UF account. The form, however, is anonymous.

Guest Lecture last Thursday: Clark Wood

For more information, please email Clark Wood (clark.wood@protonmail.com)

Last class (watch the videos from Lecture 11!)

We learned about different properties of PDFs:

1. $F_X(x) = \int_{-\infty}^{x} f_X(t)\, dt$
2. $f_X(x) \geq 0, -\infty < x < \infty$
3. $\int_{-\infty}^{\infty} f_X(t) dt = 1$
4. $P(a < X \leq b) = \int_a^b f_X(x)\, dx$ where $a \leq b$
5. If $g(x)$ is a nonnegative piecewise continuous function with finite integral $\int_{-\infty}^{\infty} g(x)\, dx = c$, then $f_X(x) = \frac{g(x)}{c}$ is a valid pdf.

We also defined the CDF and survival function for the Standard Normal RV X (Gaussian with $\mu = 0$ and $\sigma^2 = 1$), that is, $X \sim G(0, 1)$:

- **CDF of $G(0, 1)$:**

$$\Phi(x) = P(X \leq x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{t^2}{2}\right\} dt$$

- **SF of $G(0, 1)$:**

$$Q(x) = P(X > x) = \int_{x}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{t^2}{2}\right\} dt$$

We saw that:

$$\Phi(x) + Q(x) = 1, \forall x$$

The CDF and survival function for any RV $Y \sim G(\mu, \sigma^2)$ are:

$$F_X(x) = \Phi\left(\frac{x-\mu}{\sigma}\right)$$

and

$$S_X(x) = \mathrm{Q}\left(\frac{x-\mu}{\sigma}\right)$$

Note that the denominator above is σ, not σ^2. Many students use the wrong value when solving problems!

- To find the probability of some interval using the Q-function, it is easiest to rewrite the probability:

$$\begin{aligned} P(a < X \leq b) &= P(X > a) - P(X > b) \\ &= Q\left(\frac{a-\mu}{\sigma}\right) - Q\left(\frac{b-\mu}{\sigma}\right) \end{aligned}$$

- In general, Gaussian probabilities can always be expressed in terms of "tail" probabilities

More on Computing Gaussian Tail Probabilities

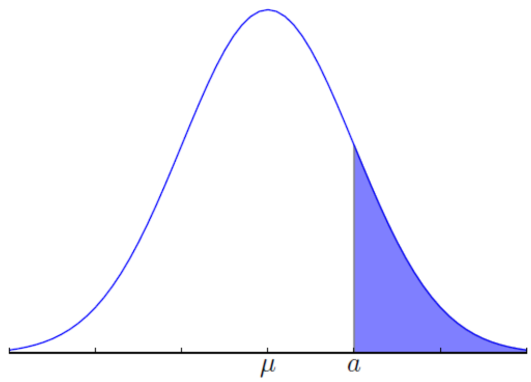
Any Gaussian probabilities can be decomposed in terms of Gaussian tail probabilities. There are 2 cases of the tail probabilities:

- **Case 1:** $P(X \geq a)$, where $a > \mu$

$$P(X \geq a) = Q\left(\frac{a-\mu}{\sigma}\right)$$

In [1]:
```
from IPython.display import Image
Image('figures/probXlargerA.png', width=400)
```

Out[1]:



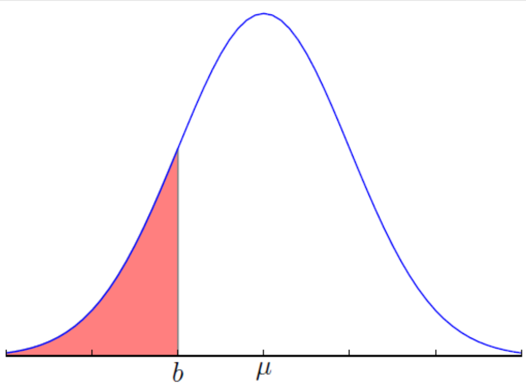
- **Case 2:** $P(X \leq b)$, where $b < \mu$

By symmetry:

$$P(X \geq \mu + (\mu - b)) = P(X \geq 2\mu - b) = Q\left(\frac{2\mu - b - \mu}{\sigma}\right) = Q\left(\frac{\mu - b}{\sigma}\right)$$

In [2]:
```
Image('figures/probXsmallerB.png', width=400)
```

Out[2]:



In [3]:
```
import scipy.stats as stats
import numpy as np
import numpy.random as npr
import matplotlib.pyplot as plt
import pandas as pd

%matplotlib inline
plt.style.use('bmh')
```

In [ ]:

Example: Grading on a curve

A professor's classroom requests that she "grades on a curve". The professor sees that the class grades can be modeled using a Gaussian distribution with parameters μ and σ^2.

Let X represent a randomly chosen student's grade.

(a) What is the probability that the student's grade is above μ?

$P(X > \mu) = Q\left(\frac{\mu-\mu}{\sigma}\right) = Q\left(\frac{0}{\sigma}\right) = \frac{1}{2}$

(b) The professor decides to use the following grading strategy:

- If the grades are more than σ above the mean, assign an A

- If the grades are within σ of the mean (μ), assign a B
- If the grades are more than σ below the mean, but less than 2σ below the mean, assign C
- If the grades are more than 2σ below the mean, but less than 3σ below the mean, assign D
- If the grades are more than 3σ below the mean, assign E

Determine the probability that a randomly chosen student gets each grade.

$$P(A) = P(X \geq \mu + \sigma) = Q\left(\frac{\mu + \sigma - \mu}{\sigma}\right) = Q(1)$$
$$P(B) = P(\mu - \sigma < X < \mu + \sigma) = 1 - 2P(X > \mu + \sigma) = 1 - 2Q\left(\frac{\mu + \sigma - \mu}{\sigma}\right) = 1 - 2Q(1)$$
$$P(C) = P(\mu - 2\sigma < X < \mu - \sigma) = P(X \geq \mu - 2\sigma) - P(X \geq \mu - \sigma) = Q(1) - Q(2)$$
$$P(D) = P(\mu - 3\sigma < X < \mu - 2\sigma) = P(X \geq \mu - 3\sigma) - P(X \geq \mu - 2\sigma) = Q(2) - Q(3)$$
$$P(E) = P(X < \mu - 3\sigma) = P(X > \mu + 3\sigma) = Q(3)$$

In [4]:
```
?stats.norm
```

In [5]:
```
def q(x):
    # by default stats.norm creates a Gaussian(0,1)
    return stats.norm.sf(x)
```

In [6]:
```
pA = q(1)
pA
```

Out[6]: 0.15865525393145707

In [7]:
```
pB = 1-2*q(1)
pB
```

Out[7]: 0.6826894921370859

In [8]:
```
pC = q(1) - q(2)

pC
```

Out[8]: 0.13590512198327787

In [9]:
```
pD = q(2) - q(3)

pD
```

Out[9]: 0.0214002339165491

In [10]:
```
pE = q(3)

pE
```

Out[10]: 0.0013498980316300933

Checking work:

In [11]:
```
pA + pB + pC + pD + pE
```

Out[11]: 1.0

(c) Suppose the threshold to get an A is $k\sigma$ above the mean, what value of k is needed for 40% of the class to get an A?

$$\begin{aligned} P(X \geq \mu + k\sigma) &= 0.4 \\ Q\left(\frac{\mu + k\sigma - \mu}{\sigma}\right) &= 0.4 \\ Q(k) &= 0.4 \\ k &= Q^{-1}(0.4) \\ k &\approx 0.25 \text{ (using Q-function table)} \end{aligned}$$

In [12]:
```
?stats.norm
```

In [13]:
```
def qinv(x):
    return stats.norm.isf(x)
```

In [14]:
```
qinv(0.4)
```

Out[14]: 0.2533471031357997

Expected Value

Consider again a set of observations $x_1, x_2, \dots, x_N$.

Then the **average** of the data is

$$\overline{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

- We would like to define a similar notion for a random variable X, but take the average over the *ensemble* of potential values of X.
- This value is the *expected value*, *ensemble mean*, or simply *mean* of X.
- We can use *relative frequency* to connect the two.

Suppose that $X = \{x_i\}_{i=1}^{n}$ are random data that take values from $S_k = \{a_1, a_2, \dots, a_k\}$, where $k < \infty$.

$$\overline{X} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

Let $n_j \equiv$ # of times a_i occurs, then

$$\overline{X} = \frac{1}{N} \sum_{j=1}^{N} a_j \cdot n_j = \sum_{j=1}^{N} a_j \cdot \frac{n_j}{N}$$

If our experiment possesses statistical regularity,

$$\lim_{n \to \infty} \frac{n_j}{N} \to p_j = P(a_j)$$

Therefore,

$$\lim_{n \to \infty} \overline{X} = \sum_{j=1}^{N} a_j \cdot p_j \triangleq E[X]$$

Expected Value The **expected value** or **mean** of a random variable X is

$$\mu_X = E[X] = \sum_x x p_X(x),$$

if X is a **discrete random variable**, and is defined as

$$\mu_X = E[X] = \int_{-\infty}^{\infty} x f_X(x)\, dx,$$

if X is a **continuous random variable**. **Note:** In some special cases, we would not define the expected value because it is of the form $-\infty, +\infty$. We won't cover those in this class.

Why do we care about the mean?

In a repeated experiment, the limit of the average value is the mean

- In fact, we will show that we can determine a limit on the number of times the experiment must be repeated to ensure that the average is within a range around the mean with a specified probability (Chebyshev's inequality, covered later)

If we wish to use a constant value to estimate a random variable, then the mean is the value that minimizes the mean-square error.

Note that $E[X]$ may be infinite.

Example 1: Rolling a fair 6-sided die.

Let X be the number of top face of die.

$$\begin{aligned} E[X] &= \sum_{i=1}^{6} i \times P(X=i) \\ &= \sum_{i=1}^{6} i \times \frac{1}{6} \\ &= \frac{21}{6} \\ &= 3.5 \end{aligned}$$

Example 2: Bernoulli RV.

Let $X \sim \text{Bernoulli}(p)$, then $p_X(x) = \begin{cases} p & x=1 \\ 1-p & x=0 \\ 0 & \text{o.w.} \end{cases}$

$$E[X] = (0)(1-p) + (1)(p) = p$$

Example 3: What is the expected value of the random variable $X \sim U(1,6)$?

Let $X \sim U(1,6)$, then $f_X(x) = \begin{cases} \frac{1}{6-1}, & x \in [1,6] \\ 0, & \text{o.w.} \end{cases} = \begin{cases} \frac{1}{5}, & x \in [1,6] \\ 0, & \text{o.w.} \end{cases}$

$$E[X] = \int_{-\infty}^{\infty} x f_X(x) dx = \int_1^6 x \frac{1}{5} dx = \frac{x^2}{10}\Big|_1^6 = \frac{6^2}{10} - \frac{1^2}{10} = \frac{35}{10} = 3.5$$

In [15]:
```
?stats.uniform
```

In [16]:
```
U = stats.uniform(loc=1,scale=5) # U(1,6)
```

In [17]:
```
# sample random values from this RV

U.rvs(size=10)
```

Out[17]:
```
array([3.58791465, 3.7062179 , 1.81192176, 5.21158107, 4.80590236,
       1.10147198, 3.41575431, 1.63224192, 4.70924999, 5.44852137])
```

In [18]:
```
U.rvs(size=1000).mean()
```

Out[18]:
```
3.5403000596360767
```

In [19]:
```
# Expected Value

U.mean()
```

Out[19]:
```
3.5
```

In [20]:
```
U.stats('m')
```

Out[20]:
```
array(3.5)
```

Example 4: What is the expected value of the random variable $X \sim \text{Exp}(\lambda)$?

Analytically:

$$E[X] = \int_0^{\infty} x\lambda e^{-\lambda x}\, dx = \lambda \int_0^{\infty} x e^{-\lambda x}\, dx$$

Need to apply integral by parts.

In [21]:
```
lam = 2

E=stats.expon(scale=1/lam)

E.mean()
```

Out[21]:
```
0.5
```

In [22]:
```
E.stats('m')
```

Out[22]:
```
array(0.5)
```

Important Properties of the Expected Value

Property 1 - Linearity

Expected value is a **linear operator**. If X and Y are random variables, and a and b are arbitrary constants, then

$$E[aX + bY] = aE[X] + bE[Y]$$

Note that this does not required that X and Y be independent.

Example 5: Expected Value of a Binomial RV.

Let B_i, $i = 1, 2, \cdots, N$ be a sequence of independent Bernoulli random variables with common parameter p. Then

$$X = \sum_{i=1}^{N} B_i$$

is a $\text{Binomial}(N, p)$ random variable.

Using the linearity property,

$$\begin{aligned} E[X] &= E\left[\sum_{i=1}^{N} B_i\right] \\ &= \sum_{i=1}^{N} E[B_i] \\ &= \sum_{i=1}^{N} p \\ &= Np \end{aligned}$$

We can also derive the same result from the PMF of a Binomial RV, but it is *way* more complicated:

$$E[X] = \sum_{n=0}^{N} n \binom{N}{n} p^n q^{N-n}$$

(Left as exercise.)

Property 2 - Expected Value of a Function

If $Y = g(X)$, it is not necessary to compute the PDF or CDF of Y to find its expected value. We can find it using X:

$$E[Y] = \int_{-\infty}^{\infty} g(x) f_X(x)\, dx$$

This is sometimes known as the **Law of the Unconscious Statistician (LOTUS)**.

Property 3 - Expected Value of a Scalar

The expected value of a scalar constant c is $E[c] = c$.

Let $g(x) = c$, then:

$$\begin{aligned} E[c] &= \int_{-\infty}^{\infty} c f_X(x)\, dx \\ &= c \int_{-\infty}^{\infty} f_X(x)\, dx \\ &= c \end{aligned}$$

- Note that $E[f(X)] \neq f(E[X])$.

Example 6: Recall that if x_i are samples drawn from a random variable X, then

$$\lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} x_i = E[X]$$

This is the result from the **Central Limit Theorem (CLT)**.

Let's create a Uniform random variable object using `scipy.stats` . Draw 10,000 sample values from it, and use the sample values to estimate $(E[U])^2$ and $E[U^2]$.

In [23]:
```
U = stats.uniform() # U(0,1)
```

In [24]:
```
N = 10_000

samples = U.rvs(size=N)
```

In [25]:
```
U.mean()
```

Out[25]: 0.5

In [26]:
```
np.mean(samples)
```

Out[26]: 0.49628760369586633

In [27]:
```
np.mean(samples)**2 # (E[U])^2
```

Out[27]: 0.24630138558218528

In [28]:
```
U.mean()**2
```

Out[28]: 0.25

In [29]:
```
np.mean(samples**2) # E[U^2] second moment!
```

Out[29]: 0.32996657891062375

In [ ]:

In [ ]:

Moments of a Random Variable

Moments The **moments** of a random variable (or of its distribution) are *expected values of powers* or related functions of the random variable. The n-th moment of a **continuous** RV X is

$$E[X^n] = \int_x t^n f_X(t)\,dt$$

The n-th moment of a **discrete** RV X is

$$E[X^n] = \sum_x x^n p_X(x)$$

* In particular, the first moment is the *mean*, $\mu_X = E[X]$.

Central Moments The **central moments** of a random variable (or of its distribution) are *expected values of mean-centered powers* or related functions of the random variable. The n-th central moment of RV X is $E[(X-\mu_X)^n]$, in general, $E[(X-E[X])^n]$. If X is a **continuous** RV, the n-th central moment is

$$E[(X-\mu_X)^n] = \int_x (t-\mu_X)^n f_X(t)\,dt$$

If X is a **discrete** RV, the n-th central moment is

$$E[(X-\mu_X)^n] = \sum_x (x-\mu_X)^n p_X(x)$$

- Moments of a random variable are expected values of the random variable raised to some power.
- For a *central moment*, the mean is subtracted from the random variable before it is raised to a power.

Because different powers spread the values of the random variable in different ways, **moments can provide additional information about a random variable other than the mean value**:

In mathematics, a moment is a specific quantitative measure of the shape of a function. The most important ones are:

1. **Mean**, the 1st moment
2. **Variance**, the 2nd central moment
3. **Skewness**, the 3rd central moment
4. **kurtosis**, the 4th central moment

Variance is the second central moment and provides a measure of how much the probability density or mass of random variable is spread away from the mean. We define it as:

$$\begin{aligned} Var[X] &= E[(X-\mu_X)^2] \\ &= E[X^2 - 2\mu_X X + \mu_X^2] \\ &= E[X^2] - 2\mu_X E[X] + \mu_X^2 \\ &= E[X^2] - 2E[X]E[X] + (E[X])^2 \\ &= E[X^2] - (E[X])^2 \end{aligned}$$

So,

$$Var[X] = E[X^2] - (E[X])^2$$

This latter formula is usually a more convenient way to find the variance.

- The variance of a Gaussian random variable is the parameter σ^2 (you can get it through integration by parts or some clever manipulation).

Properties of Variance

Let X be a random variable and b and c constant values.

1. $Var[X] = E[X^2] - (E[X])^2 \geq 0$

1. $Var[c] = 0$

1. $Var[X-c] = E[X^2] - (E[X])^2$

1. $Var[cX] = c^2 Var[X]$

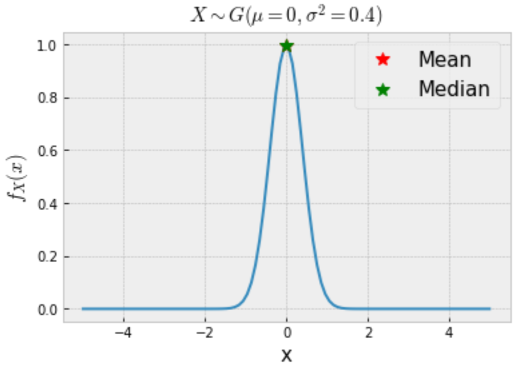
1. $Var[cX+b] = c^2 Var[X]$

We can compute the most common moments of a random variable using the `stats` module:

In [30]:
```
# Moments of G(mu=0,var=1)

G = stats.norm(loc=0,scale=1)

G.stats('mvsk')
```

Out[30]: (array(0.), array(1.), array(0.), array(0.))

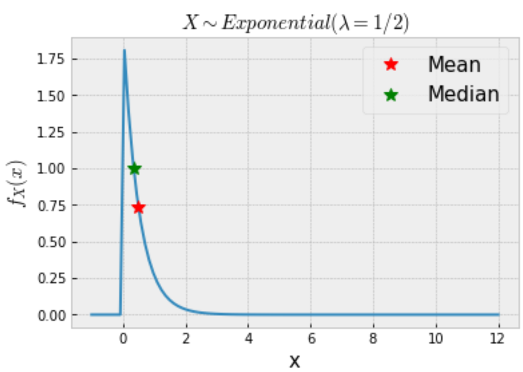
In [31]:
```
x = np.linspace(-5,5,100)
G = stats.norm(loc=0,scale=0.4) # G(0,1)

plt.plot(x,G.pdf(x))
plt.plot(G.stats('m'), G.pdf(G.stats('m')), '*r', markersize=10, label='Mean')
plt.plot(G.median(), G.pdf(G.median()), '*g', markersize=10, label='Median')
plt.legend(fontsize=15)
plt.title('$X\sim G(\mu=0,\sigma^2=0.4)$',size=15)
plt.xlabel('x',size=15)
plt.ylabel('$f_X(x)$',size=15);
```



In [32]:
```
# Moments of E(lambda=2)

lam = 2

E = stats.expon(scale = 1/lam)

E.stats('mvsk')
```

Out[32]: (array(0.5), array(0.25), array(2.), array(6.))

In [33]:
```
x = np.linspace(-1,12,100)
plt.plot(x,E.pdf(x))
plt.plot(E.stats('m'), E.pdf(E.stats('m')), '*r', markersize=10, label='Mean')
plt.plot(E.median(), E.pdf(E.median()), '*g', markersize=10, label='Median')
plt.legend(fontsize=15)
plt.title('$X\sim Exponential(\lambda=1/2)$',size=15)
plt.xlabel('x',size=15)
plt.ylabel('$f_X(x)$',size=15);
```



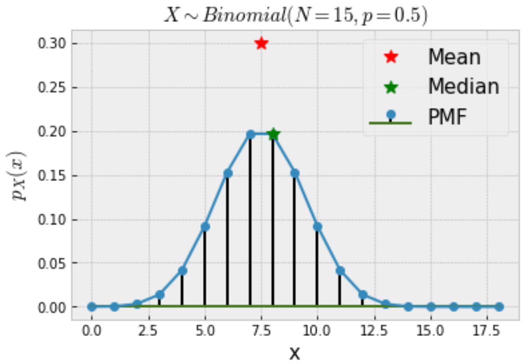
In [34]:
```
# Moments of Binomial(15,0.5)

Bn = stats.binom(15, 0.5)

Bn.stats('mvsk')
```

Out[34]: (array(7.5), array(3.75), array(0.), array(-0.13333333))

In [35]:
```
Bn.median()
```

Out[35]: 8.0

In [36]:
```
x = range(19)
plt.stem(x,Bn.pmf(x),'k',label='PMF')
plt.plot(x, Bn.pmf(x))
plt.plot(Bn.stats('m'), 0.3, '*r', markersize=10, label='Mean')
plt.plot(Bn.median(), Bn.pmf(Bn.median()), '*g', markersize=10, label='Median')
plt.legend(fontsize=15)
plt.title('$X\sim Binomial(N=15,p=0.5)$',size=15)
plt.xlabel('x',size=15)
plt.ylabel('$p_X(x)$',size=15);
```



In [ ]:

Distribution of Binary Hypothesis

In binary hypothesis testing, we built statistics that access a specific moment of the distribution (mean-difference, median-difference, etc.). We can also consider a full distribution for our hypotheses.

That is, we will be conditioning random variables depending on which hypothesis is true: $f_X(x|H_0)$ or $f_X(x|H_1)$.

In general, there are multiple ways in which random variables can depend on events or on other random variables.

We consider the same type as above: there is dependence between a random variable and some event, such that the distribution of the random variable is known if the event is known.

Motivating Case Study: Binary Communications

In a binary communication system, the received signal is a noisy version of the transmitted signal.

In the presence of thermal noise, the received signal can be modeled as $X = s_i + N$, where:

- $s_i \in \{-1, 1\}$ depends on which signal is transmitted (-1 or 1), and
- N is a Gaussian random variable with mean 0 and variance σ^2, which determines the signal-to-noise ratio

Thus, the received signal has a conditional distribution, depending on which signal is transmitted:

$$\begin{cases} X \sim \text{Gaussian}(+1, \sigma^2), & 1 \text{ transmitted} \\ X \sim \text{Gaussian}(-1, \sigma^2), & -1 \text{ transmitted} \end{cases}$$

(Here only the mean changes and not the variance, but this is an accurate model of what happens in most binary communication systems.)

We can write the conditional density and distribution functions given that i was transmitted as $f_X(x|i \sim T_x)$ and $F_X(x|i \sim T_x)$.

Example 7 Let T_i denote the event that i is transmitted. Calculate the probability that $X > 2$ if $P(T_1) = 0.25$, $P(T_{-1}) = 0.75$ and $\sigma^2 = 4$.

We can easily solve this problem using what we already know about conditional probability and Gaussian random variables:

$$\begin{aligned} P(X > 2) &= P(X > 2|T_1)P(T_1) + P(X > 2|T_{-1})P(T_{-1}) \text{ , by Total Probability} \\ &= Q\left(\frac{2-1}{\sigma}\right)(0.25) + Q\left(\frac{2-(-1)}{\sigma}\right)(0.75) \\ &= Q\left(\frac{1}{2}\right)(0.25) + Q\left(\frac{3}{2}\right)(0.75) \\ &\approx 0.127 \end{aligned}$$

In [37]:

```
stats.norm.sf(0.5)*0.25 + stats.norm.sf(3/2)*0.75
```

Out[37]:

```
0.12723978563314026
```

Example 8 Calculate the probability of error if the decision rule is:

$$\begin{cases} \hat{T}_1, & X \geq 0 \\ \hat{T}_{-1}, & X < 0, \end{cases}$$

where $\hat{T}_i$ denotes deciding that i was transmitted. Provide a numerical answer when $\sigma^2 = 0.1$ and $P(T_{-1}) = P(T_1) = \frac{1}{2}$.

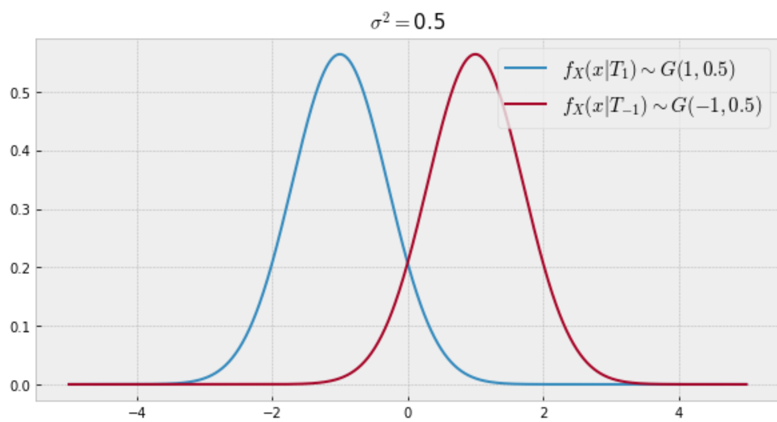
$$\begin{aligned} P(E) &= P(X \geq 0 \cap T_{-1}) + P(X < 0 \cap T_1) \\ &= P(X \geq 0 | T_{-1}) P(T_1) + P(X < 0 | T_1) P(T_1) \\ &= Q\left(\frac{0-(-1)}{\sigma}\right) P(T_{-1}) + Q\left(\frac{1-0}{\sigma}\right) P(T_1) \\ &= Q\left(\frac{1}{\sigma}\right) \\ &\approx 0.00078 \end{aligned}$$

In [38]:
```
stats.norm.sf(1/np.sqrt(0.1))
```

Out[38]: `0.000782701129001274`

What if the variance of the Gaussian noise was larger?

In [39]:
```
# Define sigma_sq
sigma_sq = 0.5


# Define normal distributions with variance sigma_sq
G_Tminus1 = stats.norm(loc=1,scale=np.sqrt(sigma_sq))
G_Tplus1 = stats.norm(loc=-1,scale=np.sqrt(sigma_sq))

x = np.linspace(-5,5,1000)

plt.figure(figsize=(10,5))
plt.plot(x, G_Tplus1.pdf(x), label='$f_X(x|T_{1})\sim G(1,'+str(sigma_sq)+')$')
plt.plot(x, G_Tminus1.pdf(x), label='$f_X(x|T_{-1})\sim G(-1,'+str(sigma_sq)+')$')
plt.legend(fontsize=15)
plt.title('$\sigma^2=$'+str(sigma_sq), size=15);
```



Optimal Decisions

Now suppose that we want to make an optimal decision $\hat{T}_i$ based on observing the value of X at the output of the receiver. I.e., given that we have $X = x$ for some value x.

The MAP rule is to choose the most probable value that was transmitted given the observation. So, that corresponds to:

- If $P(T_0|X = x) \geq P(T_1|X = x)$, decide 0
- If $P(T_1|X = x) > P(T_0|X = x)$, decide 1

and by the definition of conditional probability

$$P(T_0|X = x) = \frac{P(T_0 \cap X = x)}{P(X = x)}$$

But both the numerator and denominator are 0, because X is a continuous random variable!

Point Conditioning

Suppose we want to evaluate the probability of an event A given that $X = x$, where X is a continuous random variable.

- Then, if we use the definition of conditional probability

$$P(A|X = x) = \frac{P(A \cap X = x)}{P(X = x)} = \frac{0}{0},$$

This case is called ***point conditioning***, and we treat it as a special case:

$$\begin{aligned} P(A|X=x) &= \lim_{\Delta x \to 0} P(A|x < X \leq x + \Delta x) \\ &= \lim_{\Delta x \to 0} \frac{F_X(x+\Delta x|A) - F_X(x|A)}{F_X(x+\Delta x) - F_X(x)} P(A) \\ &= \lim_{\Delta x \to 0} \frac{\frac{F_X(x+\Delta x|A) - F_X(x|A)}{\Delta x}}{\frac{F_X(x+\Delta x) - F_X(x)}{\Delta x}} P(A) \\ &= \frac{f_X(x|A)}{f_X(x)} P(A), \end{aligned}$$

if $f_X(x|A)$ and $f_X(x)$ exist, and $f_X(x) \neq 0$.

(The result looks like what you would do if you didn't know any better -- treat the densities as if they were probabilities, and everything works out!)

Point Conditioning Form of Conditional Probability

$$P(A|X=x) = \frac{f_X(x|A)P(A)}{f_X(x)}$$

Implications of Point Conditioning

Note that the form above is almost a Bayes' rule form: If A is some input event and $f_X(x|A)$ is the likelihood of X given A, then $P(A|X=x)$ is the *a posteriori* probability of A given that the output of the system is $X=x$.

However, we don't know how to calculate $f_X(x)$ yet.

Total Probability for Conditional Density Functions

If $\{A_i\}$ form a partition of S, then from our previous work on the Law of Total Probability, we have

$$\begin{aligned} F_X(x) &= P(X \leq x) \\ &= \sum_i P(X \leq x|A_i)P(A_i) \\ &= \sum_i F_X(x|A_i)P(A_i) \end{aligned}$$

Since $f_X(x) = \frac{d}{dx}F_X(x)$ and $f_X(x|A_i) = \frac{d}{dx}F_X(x|A_i)$,

$$\begin{aligned} f_X(x) &= \frac{d}{dx}F_X(x) \\ &= \frac{d}{dx}\sum_i F_X(x|A_i)P(A_i) \\ &= \sum_i \left[\frac{d}{dx}F_X(x|A_i)\right] P(A_i) \\ &= \sum_i f_X(x|A_i)P(A_i) \end{aligned}$$

Continuous Version of Law of Total Probability

$$P(A|X=x) = \frac{f_X(x|A)}{f_X(x)}P(A)$$

$$\Rightarrow P(A|X=x)f_X(x) = f_X(x|A)P(A)$$

$$\Rightarrow \int_{-\infty}^{\infty} P(A|X=x)f_X(x)dx = \int_{-\infty}^{\infty} f_X(x|A)dxP(A)$$

$$\Rightarrow P(A) = \int_{-\infty}^{\infty} P(A|X=x)f_X(x)dx$$

Point Conditioning Form of the Law of Total Probability If $\{A_i,\ i=1,2,\ldots,n\}$ form a partition of S, then

$$F_X(x) = \sum_{i=1}^{n} F_X(x|A_i)P(A_i)$$

Since $f_X(x) = \frac{d}{dx}F_X(x)$ and $f_X(x|A_i) = \frac{d}{dx}F_X(x|A_i)$, we also have

$$f_X(x) = \sum_{i=1}^{n} f_X(x|A_i)P(A_i)$$

Point Conditioning Form of Bayes' Rule

If $\{A_i,\ i = 1, 2, \ldots, n\}$ form a partition of S, then

$$P(A_i|X = x) = \frac{f_X(x|A_i)P(A_i)}{\sum_{i=1}^{n} f_X(x|A_i)P(A_i)}$$

P C F o B R

If { fc a p o S, th F

Example 9

Find the MAP decision rules for the binary communication system.

The MAP rule is to choose $\hat{T}_1$ if T_1 maximize

$$P(T_1|X = x)$$

$$\frac{f_X(x|T_1)P(T_1)}{f_X(x)} \geq$$

$$f_X(x|T_1)P(T_1) \geq$$

and choose $\hat{T}_{-1}$ otherwise

We coulc simpl this furth by subs in the cond dens but plotti the weigl dens and the decis regio will give us much more insigl

In [57]:

```
def drawMAP(p,sigma2=1):
    ''' Draw the weighted densities for the binary communication sy
    and shade under them according to the MAP decision rule.

    Inputs:
    p0= probability that 0 is transmitted
    sigma2= variance of the Gaussian noise (default is 1)'''

    # Set up random variables
    x = np.linspace(-4,4,1000)
    Gplus1 = stats.norm(loc = 1, scale = np.sqrt(sigma2)) #data lik
    Gminus1 = stats.norm(loc = -1, scale = np.sqrt(sigma2)) # data

    # Prior probabilities
    # p is given, the probability of sending a 1
    p_minus1 = 1-p


    # plot the weighted densities:
    # these are proportional to the APPs
    plt.figure(figsize=(10,5))
    plt.plot(x, p*Gplus1.pdf(x), label='$f_X(x|T_1)P(T_1)$')
    plt.plot(x, p_minus1*Gminus1.pdf(x), label='$f_X(x|T_{-1})P(T_{


    # Determine the regions where the APP for 1 is
    # bigger and the APP for -1 is bigger
    Rplus1 = x[p*Gplus1.pdf(x)  >= p_minus1*Gminus1.pdf(x)] # regic
    Rminus1 = x[p_minus1*Gminus1.pdf(x) > p*Gplus1.pdf(x)] # region


    # Fill under the regions found above
    plt.fill_between(Rplus1, p*Gplus1.pdf(Rplus1), alpha = 0.3, lab
    plt.fill_between(Rminus1, p_minus1*Gminus1.pdf(Rminus1), alpha

    plt.xlabel('$x$',size=15); plt.ylabel('Weigthed PDF\n$f_X(x|T_i
    plt.legend(fontsize=15)

    # Print the MAP threshold
    print('MAP decision threshold to decide T1 is >', round(Rplus1[
```