i.e. what type of distribution

Categorization learning with non-parametric method

Assume that we know the functional form of the response time (RT) distribution for normal monosyllabic words. By having this assumption, we make our inference task easy: instead of inferring the entire RT distribution, we simply have to infer one parameter (in this case, μ) in the distribution for this category. Therefore, this method is also known as a parametric method.

mean

However, in reality, we may not know the exact functional form of the category distribution (let alone the parameters in the function). In this tutorial, we are going to look at a method in which we can estimate the entire category distribution without knowing the exact functional form of the category distribution, which is also known as **a non-parametric method for categorization.**

How??

Intuitively, every time you observe the RT of a new normal monosyllabic word, you know a little more about how the underlying distribution is like. How do we add up these bits of knowledge across observations? We do this by proposing a "kernel" function, which captures the "knowledge" from each observation.

Suppose we've got RT measurements from N normal words, and we're considering M possible values in the "RT space" (e.g., if we consider RTs from 1 to 1000, with stepsize = 0.1).

Let's start with a Gaussian kernel, which has the following form:

N words $\subseteq$ M values

$$k(t_j, y_i) = e^{\left(\frac{-(t_j - y_i)^2}{2h^2}\right)}$$

y_i is the RT for observation i, for $i = 1,2,\ldots,N$, and t_j is the j^{th} value in the RT space we consider for the normal word RT distribution, where $j = 1,2,\ldots,10000$ in this simulation.

h is a window parameter that controls the width of the Gaussian kernel. We'll see what this means later.

For now, let's set $N = 50$ as we will have 50 exemplar observations, and $h = 20$.

For the RT space, we make a vector `T` = 1:stepsize:1000, where stepsize = 0.1.

RTNorm.mat contains a vector, `NormRT`, which store the RT data for 50 normal words. Load this MAT file at the beginning of your script.

In-Class Exercise

1. For each observed RT, compute the kernel values for all the possible values in the RT space `T`.

2. Sum up the ~~all~~ kernel values across all observations. The resulting sum is now proportional to the target RT distribution that we want to infer. Formally, if T represents all possible RT values,

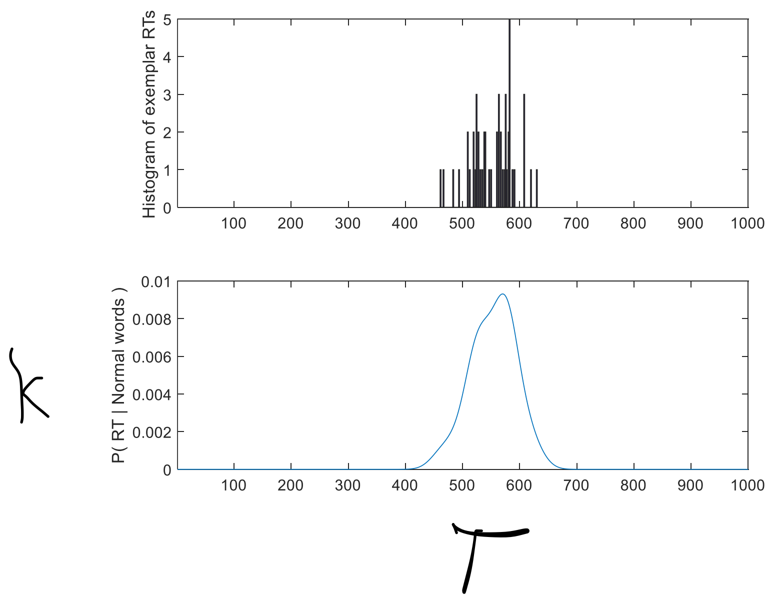
$$\sum_{i=1}^{N} k(T, y_i) = K(T) \propto P(T \mid \text{normal monosyllabic words})$$

3. Normalize $K(T)$, so that it becomes a probability density function (PDF), i.e.,

$$\int_{-\infty}^{+\infty} K(t)dt = 1 \quad \Rightarrow \text{ approximated as } \Rightarrow \quad \sum_{j=1}^{M} K(t_j)\Delta t = 1$$

4. Compute the mean and SD of your inferred PDF.
   [Option 1: use the method we talked about in the previous tutorial with the equation below]
   $$E(\mu) = \int_{-\infty}^{+\infty} t\, P(\mu = t)dt \;\approx\; \sum_{i=1}^{N} \mu_i P(\mu = \mu_i)\, \Delta\mu$$
   $Var(X) = \int_{-\infty}^{+\infty} [x - E(X)]^2\, P(X = x)dx]$.

   [Option 2: use sampling method with the calculated density distribution K(T), and then compute sample mean and sample variance from the samples].




```
M =

  553.3413


Var =

   1.8109e+03
```

5. Use the same set of code on the RT data for 50 **taboo words** (with the input file RTTaboo.mat). What distribution do you see? Write a comment at the end of your Matlab code for this question.