# Homework 4

Solutions

1. (Half life)

   (a) We have

$$
\begin{aligned}
\mathrm{P}(\tilde{t} > t_{1/2}) &= \int_{t_{1/2}}^{\infty} \lambda \exp\left(-\lambda x\right) \,\mathrm{d}x && (1)\\
&= \exp\left(-\lambda t_{1/2}\right). && (2)
\end{aligned}
$$

   Setting equal to 1/2 yields $t_{1/2} = \frac{\ln 2}{\lambda}$. This is the time that it takes for the particle to decay with probability 1/2. This seems like a reasonable definition, because if e have a group of particles all following the same distribution, we can consider each as a realization of an experiment where we check whether a particle survives after $t_{1/2}$. By our *intuitive* definition of the probability of an event, about half of them will survive because the probability of the event is 1/2.

   (b) We have

$$
\begin{aligned}
\mathrm{P}(t_{1/2} < \tilde{t} < t) &= \int_{t_{1/2}}^{t} \lambda \exp\left(-\lambda x\right) \,\mathrm{d}x && (3)\\
&= \exp\left(-\lambda t_{1/2}\right) - \exp\left(-\lambda t\right) && (4)\\
&= \frac{1}{2} - \exp\left(-\lambda t\right). && (5)
\end{aligned}
$$

   Setting equal to 1/4 yields $t_{1/4} = \frac{\ln 4}{\lambda} = \frac{2\ln 2}{\lambda} = 2t_{1/2}$. Intuitively, this means that after half the particles have decayed, it takes the same time for one quarter of the particles to decay. This is consistent with the intuitive definition of half life because once half of the particles are gone, half of the remaining ones (i.e. one quarter) should decay after the half time.

   (c)

$$
\begin{aligned}
\mathrm{P}(\tilde{t} > k t_{1/2}) &= \int_{k t_{1/2}}^{\infty} \lambda \exp\left(-\lambda x\right) \,\mathrm{d}x && (6)\\
&= \exp\left(-\lambda k t_{1/2}\right) && (7)\\
&= \exp\left(-\ln 2^k\right) && (8)\\
&= \frac{1}{2^k}. && (9)
\end{aligned}
$$

   This says that if we wait for $k$ half times, the remaining particles are halved $k$ times, which is consistent with the intuitive definition of half time.

2. (Triangular pdf)

   (a) The possible values of $w$ are $w \geq \max\left(x_1, \ldots, x_n\right) = 1.5$.

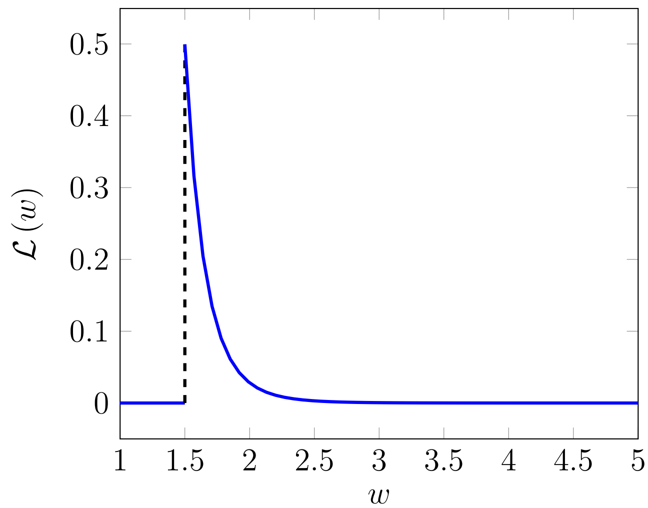
(b) The likelihood equals

$$\mathcal{L}(w) = \prod_{i}^{n} f_w(x_i) \tag{10}$$

$$= \frac{2^5}{w^{10}} \prod_{i}^{n} x_i \tag{11}$$

$$= \frac{28.8}{w^{10}} \tag{12}$$

if $w \geq \max(x_1, \ldots, x_n)$ and 0 otherwise because in that case $f_w(\max(x_1, \ldots, x_n)) = 0$.




(c) The maximum likelihood estimate is 1.5 because the likelihood is decreasing over $[1.5, \infty)$.

(d) The ML estimate systematically underestimates the true parameter. It is equal to the maximum value in the data, which has to be smaller than $w_{\text{true}}$.

(e) The cdf equals

$$F_w(x) = \begin{cases} 0, & x \leq 0, \\ \frac{x^2}{w^2}, & \text{for } 0 \leq x \leq w, \\ 1, & \text{for } x \geq w. \end{cases} \tag{13}$$

The inverse of the cdf in the interval of interest is $w\sqrt{y}$. The sample is $2\sqrt{0.64} = 1.6$.

3. (Planet)

(a) We have

$$F_{\tilde{t}}(t) = \int_{-\infty}^{t} f_{\tilde{t}}(u)\, \mathrm{d}u. \tag{14}$$

For $t < 0$, this yields

$$F_{\tilde{t}}(t) = \int_{-\infty}^{t} \frac{\lambda \exp(\lambda u)}{2} \, \mathrm{d}u \tag{15}$$

$$= \frac{\exp(\lambda t)}{2}. \tag{16}$$

For $t \geq 0$,

$$F_{\tilde{t}}(t) = \int_{-\infty}^{0} \frac{\lambda \exp(\lambda u)}{2} \, \mathrm{d}u + \int_{0}^{t} \frac{\lambda \exp(-\lambda u)}{2} \, \mathrm{d}u \tag{17}$$

$$= \frac{1}{2} + \frac{1 - \exp(-\lambda t)}{2} \tag{18}$$

$$= 1 - \frac{\exp(-\lambda t)}{2}. \tag{19}$$

(b) The log-likelihood is

$$\log \mathcal{L}_X(\lambda) = \sum_{i=1}^{n} \log f_\lambda(x_i) \tag{20}$$

$$= \sum_{i=1}^{n} \log \frac{\lambda \exp(-\lambda |x_i|)}{2} \tag{21}$$

$$= n \log \lambda - n \log 2 - \lambda \sum_{i=1}^{n} |x_i| \, . \tag{22}$$

The derivative and second derivative of the log-likelihood function are given by

$$\frac{\mathrm{d} \log \mathcal{L}_{x_1,\ldots,x_n}(\lambda)}{\mathrm{d}\lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} |x_i| \, , \tag{23}$$

$$\frac{\mathrm{d}^2 \log \mathcal{L}_{x_1,\ldots,x_n}(\lambda)}{\mathrm{d}\lambda^2} = -\frac{n}{\lambda^2} < 0 \quad \text{for all } \lambda > 0. \tag{24}$$

The function is concave, as the second derivative is negative, so there cannot be different local maxima. The maximum is obtained by setting the first derivative to zero, which yields

$$\lambda_{\mathrm{ML}} = \frac{1}{\frac{1}{n}\sum_{i=1}^{n} |x_i|} \tag{25}$$

$$= \frac{1}{39}. \tag{26}$$

(c) The conditional cdf of $\tilde{t}$ given $\tilde{t} \geq 0$ evaluated at $t > 0$ is

$$
\begin{aligned}
F_{\tilde{t}\,|\,\tilde{t}>0}(t) &= \mathrm{P}\left(\tilde{t} \leq t \,|\, \tilde{t} > 0\right) && (27)\\
&= \frac{\mathrm{P}\left(0 < \tilde{t} \leq t\right)}{\mathrm{P}\left(\tilde{t} > 0\right)} && (28)\\
&= \frac{F_{\tilde{t}}(t) - F_{\tilde{0}}}{1 - F_{\tilde{t}}(0)} && (29)\\
&= \frac{1 - \frac{\exp(-\lambda t)}{2} - \frac{1}{2}}{\frac{1}{2}} && (30)\\
&= 1 - \exp(-\lambda t). && (31)
\end{aligned}
$$

Differentiating with respect to $t$ yields an exponential pdf $f_{\tilde{t}|\tilde{t}>t_0}(t) = \lambda e^{-\lambda t}$ with parameter $\lambda$.

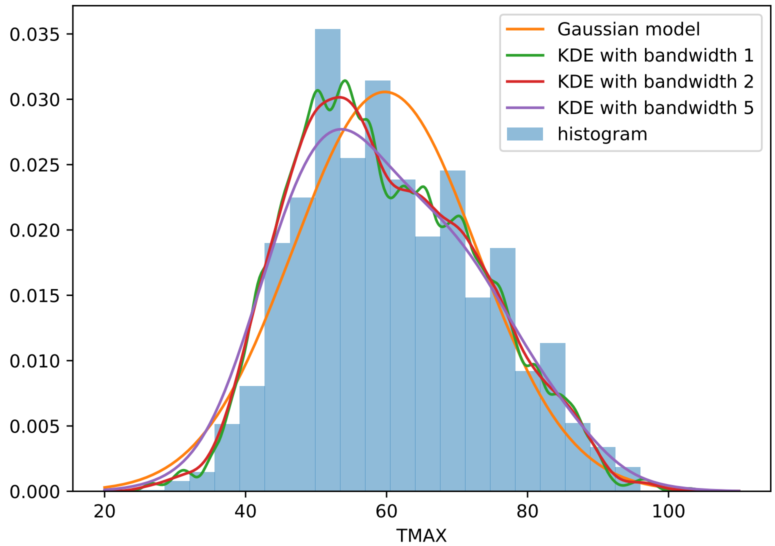
4. (a)

```
train_df = pd.read_csv('weather_train.csv')
test_df = pd.read_csv('weather_test.csv')

# Gaussian model
def get_parametric_model(x, data):
    return scipy.stats.norm.pdf(x, data.mean(), data.std())

# KDE
def get_kde_model(x, data, bandwidth):
    num_samples = len(data)
    x_tmp = (x[:, np.newaxis] - np.array(data)) / bandwidth
    weighted_kernels = scipy.stats.norm.pdf(x_tmp, 0, 1).sum(axis=1)
    return weighted_kernels / (num_samples * bandwidth)

x = np.linspace(20, 110, 1000)
fig, ax1 = plt.subplots(1,1, figsize=(7,5))
ax1.hist(test_df['TMAX'], bins=20, alpha=0.5, \
                label='histogram', density=True)
ax1.plot(x, get_parametric_model(x, train_df['TMAX']), \
                label="Gaussian model")
for h in [1,2,5]:
    ax1.plot(x, get_kde_model(x, train_df['TMAX'], bandwidth=h), \
                label="KDE with bandwidth {}".format(h))
ax1.set_xlabel('TMAX')
plt.legend()
plt.show()
```




KDE (with bandwidth 2) visually performs better.

(b) Gaussian model with MLE performs slightly better. In (a), the data is skewed, so KDE is more robust to fit the data. However, when the data has less samples and closer to Gaussian, Gaussian distribution wiht MLE can avoid overfitting.