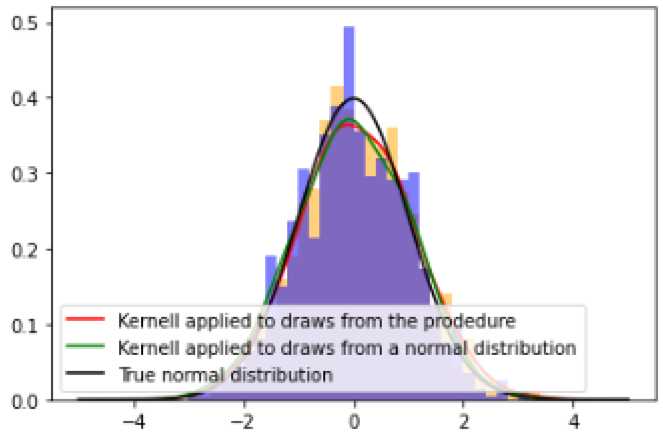
Machine Learning

Homework #(**2**)
JARRY Guillaume

1 Q1

The following code generate the figure below:

```python
import numpy as np
import matplotlib.pyplot as plt

def gaussian_estimate():
    S = 0
    for i in range(12):
        S += np.random.uniform()
    return S - 6

normal_sample = np.random.randn(1000)
estimate_sample = np.asarray([gaussian_estimate() for _ in range(1000)])
x = np.linspace(-5, 5, 1000)

def Gaussian_Kernel_density(x, X, h=0.35):
    return sum( [1/np.sqrt(2*np.pi*h**2)*np.exp(-(x - X[i])**2/(2*h**2)) for i in range(X.shape[0])]) / X.shape[0]

Kernell_estimate = Gaussian_Kernel_density(x, estimate_sample)
Kernell_sample = Gaussian_Kernel_density(x, normal_sample)

def gaussian(x):
    return 1/np.sqrt(2*np.pi)*np.exp(-(x)**2/(2))

Gaussian = gaussian(x)

plt.plot(x, Kernell_estimate, label='Kernell applied to draws from the prodedure', color='red')
plt.plot(x, Kernell_sample, label='Kernell applied to draws from a normal distribution', color='green')
plt.hist(estimate_sample, bins=50, density=True, range=(-5, 5), color='orange', alpha=0.5)
plt.hist(normal_sample, bins=50, density=True, range=(-5, 5), color='blue', alpha=0.5)
plt.plot(x, Gaussian, label='True normal distribution', color='black')
plt.legend(loc='lower left')
plt.show()
```



We can see with the plotting that the standard deviation of the gaussin kernell applied to our sample actually dictates how close we want to stay to the sample. Thus a very small standard deviation will pick up on a lot of noise from the original signal, which is not what we are trying to do here. Thus a higher standard deviation will have a smoothing effect appropriate here to get closer to the original gaussian distribution. The downside is that the higher the standard deviation, the flatter it will estimate the original distribution. This comes from the fact that with a higher std, when the Kernell estimate is computed, the datapoint farther from the estimate matters more, thus having a flattening effect.

Machine Learning

Homework #(**2**)
JARRY Guillaume

---

2 Q2

1) a) The pdf of a Poisson law is of the form $f(x,\lambda) = \frac{\lambda^x}{x!}e^{-\lambda}$ where x is an integer and $\in [0,1]$ is the parameter. If we pose $\eta = log(\lambda)$ we can easily recognize an exponential family where:

$$h(x) = x!$$
$$\psi(x) = x$$
$$\eta = log(\lambda)$$
$$A(\eta) = e^{\eta}$$

b) The pdf of a multivariate normal distribution is of the form:

$$f(x,\mu,I) = f(x,\mu) = \frac{1}{(2\pi)^d} exp(-\frac{1}{2}\|X-\mu\|)^2 = \frac{1}{(2\pi)^d} exp(-\frac{1}{2}(X^TX - 2\mu^TX + \mu^T\mu))$$

Thus by posing:

$$h(x) = \frac{1}{(2\pi)^d}$$
$$\psi(x) = \begin{bmatrix} X^TX \\ X \end{bmatrix}$$
$$\eta = \begin{bmatrix} -\frac{1}{2} \\ \mu \end{bmatrix}$$
$$A(\eta) = \eta^T \begin{pmatrix} 0 & 0 \\ 0 & I_d \end{pmatrix} \eta$$

We can recognize the exponential family. Beware that η and $\psi(x)$ is of dimension $d+1$.
c) The pdf of a multinomial distribution is of the form:

$$f(n,\theta) = \frac{n!}{(n-\sum_{k=1}^{K-1} n_k)! \times \prod_{k=1}^{K-1} n_k!}(1-\sum_{k=1}^{K-1}\theta_k)^{n-\sum_{k=1}^{K-1} n_k}\prod_{k=1}^{K-1}\theta_k^{n_k}$$

We can reduce the dimension of the input n because of the dependace of n_K on the other coordinates, thus we recognize the exponential family by posing:

$$x = \begin{bmatrix} n_1 \\ \vdots \\ n_{K-1} \end{bmatrix}$$
$$h(x) = \frac{(1^Tx)!}{(n-1^Tx)! \times \prod_{k=1}^{K-1} n_k}$$
$$\psi(x) = x$$
$$\eta = log(\frac{\theta'}{1-1^T\theta'}) \text{ where } \theta' = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_{K-1} \end{bmatrix}$$

$$A(\eta) = n \times log\left(1 + 1^T e^{\eta}\right)$$

Where it is important to note that here the function log and exp are applied to the vector element wise.

2) We have the property of the log partition function that:

$$A(\eta) = log(\int_{\Xi} h(x)exp(\eta^T \psi(x))dx)$$

(This comes from the fact that the integral of the pdf must be equal to 1.

Thus, when we apply the gradient, according to the composition of gradient rule and the derivation under the sum rule, we will get the vector:

$$\nabla_{\eta} A(\eta) = \left(\frac{\int_{\Xi} h(x)\psi_i(x)exp(\eta^T \psi(x))dx}{\int_{\Xi} h(x)exp(\eta^T \psi(x))dx}\right)_{1 \leq i \leq dim\eta}$$

And then, because the denominator is a constant equal to $exp(A(\eta))$ and integration is linear, we end up with the expression.

$$\nabla_{\eta} A(\eta) = \left(\int_{\Xi} h(x)\psi_i(x)exp(\eta^T \psi(x) - A(\eta))dx\right)_{1 \leq i \leq dim\eta}$$

And then by simplifying the integral we can recognize an expectation formula:

$$\nabla_{\eta} A(\eta) = \int_{\Xi} h(x)\psi(x)exp(\eta^T \psi(x) - A(\eta))dx = E(\psi(x))$$

3) Exploiting the relationship found in the previous question, we get:

a) Poisson's Law:

$$\nabla_{\eta} A(\eta) = \eta e^{\eta} = log(\lambda) = E(x!))$$

b) Multivariate normal distribution:

$$\nabla_{\eta} A(\eta) = \begin{pmatrix} 0 & 0 \\ 0 & 2I_d \end{pmatrix} \eta = (0, 2\mu)$$

Thus, $E(\psi(x)) = (0, 2\mu)$

c) Mutlinomial distribution :

$$\nabla_{\eta} A(\eta) = n \times \frac{e^{\eta}}{1 + 1^T e^{\eta}} = n\theta = E(x)$$

We are happy to find the expected results.

3 Q3

1) Using Bayes' Rule, we get :

$$p(\theta|X) = \frac{f(X|\theta)f(\theta)}{p(X)}$$

Now,$p(X)$ is a constant and therefore does not intervene in the calculation of the maximizer. Then, because all the x_i are independantly distributed:

$$f(X|\theta) = \prod_{i=1}^{n} f(x_i|\theta)$$

Thus, the MAP is going to be:

$$MAP(\theta) = \frac{1}{n} log\left(\frac{f(\theta)\prod_{i=1}^{n} f(x_i|\theta)}{p(X)}\right)$$
$$= \frac{1}{n}\left(-\frac{\|\theta-\theta_0\|^2}{2\tau^2} - \sum_{i=1}^{n}\frac{\|x_i-\theta\|^2}{2\sigma^2} - log(p(X))\right)$$

Then, because $p(X)$ is fixed, it does not intervene in the computation of the maximzers, and so we get:

$$\hat{\theta}_{MAP} = \underset{\theta}{agrmax}\left(\frac{1}{n}\left(-\frac{\|\theta-\theta_0\|^2}{2\tau^2} - \sum_{i=1}^{n}\frac{\|x_i-\theta\|^2}{2\sigma^2}\right)\right)$$

Here we recognize a least square problem: The objective of the problem is to find θ which minimize the distance between all the X_i and θ_0. The answer is going to be the barycenter of all these points. And thus we get :

$$\hat{\theta}_{MAP} = \frac{1}{(N+1)\sigma^2}\sum_{i=1}^{n} X_i + \frac{\theta_0}{(N+1)\tau^2}$$

2) Because the X_i are independant from one another, the likelihood estimate of θ is going to be:

$$likelihood(\theta, X) = \prod_{i=1}^{n} f(x_i|\theta)$$

And thus the log-likelihood will be:

$$-l(\theta, X) = nlog(\sigma\sqrt{2\pi}) + \sum_{i=1}^{n}\frac{\|x_i-\theta\|^2}{2\sigma^2}$$

Because σ is fixed, we already know the solution to this problem. The minimum of the log-likelihood is going to be the barycenter of all the X_i with $i = 1, \ldots, N$ and thus:

$$\hat{\theta}_{MLE} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

3) When n goes to infinty, the term in θ_0 goes to zero and according to the laws of large numbers :

$$\frac{1}{(N+1)\sigma^2}\sum_{i=1}^{n} X_i \underset{N\to\infty}{\sim} \frac{1}{N\sigma^2}\sum_{i=1}^{n} X_i \underset{N\to\infty}{\sim} E(X_i) = \theta$$

The MAP estimator converges well and has no bias.

$$MAP(\theta) \underset{n\to\infty}{\to} \theta$$

Machine Learning

Homework #(**2**)
JARRY Guillaume

---

4 Q4

First let us remark that the function $x \to -xlog(x)$ is concave on R. It is clearly the case because its second order derivative $x \to -\frac{1}{x^2}$ is negative. Thus we easily get the left inequality:

$$\forall i \in \{1, \ldots, n\}, -\lambda p_i log(p_i) - (1-\lambda)q_i log(q_i) \leq -(\lambda p_i + (1-\lambda)q_i)log(\lambda p_i + (1-\lambda)q_i)$$

And by summation of positive terms:

$$\lambda H(X) + (1-\lambda)H(Y) \leq H(Z_\lambda)$$

The right inequality is a bit trickier. Let us introduce B a Bernoulli variable of parameter λ and then we will define a variable

$$T = \begin{cases} P \text{ if } B = 1 \\ Q \text{ if } B = 0 \end{cases}$$

Then we recognize, according to the law of total probability:

$$P(Z_\lambda) = \lambda P(T|B=1) + (1-\lambda)P(T|B=0)$$

Then, we have the following inequality on conditional entropy (*):

$$H(Z_\lambda) = H(T, B) \leq H(T|B) + h(\lambda)$$

Where here, $H(B) = h(\lambda)$ and, using the laws of total probablity shown earlier:

$$H(T|B) = \lambda H(X) + (1-\lambda)H(Y)$$

And thus we obtain the right inequality :

$$H(Z_\lambda) \leq \lambda H(X) + (1-\lambda)H(Y) + H(B)$$

(*) This inequality was found n this paper, page 3: Estimating Mixture Entropy with Pairwise Distances Artemy Kolchinsky and Brendan D. Tracey