1

a)

The definition of total variance can be spitted into two cases. In particular:

$$d_{TV} = max_{S\subseteq[n]}|p(S) - q(S)| = \begin{cases} max_{S\subseteq[n]}(p(S) - q(S)) & \text{if } (p(S) - q(S)) \geq 0 \\ max_{S\subseteq[n]}(q(S) - p(S)) & \text{if } (p(S) - q(S)) < 0 \end{cases}$$

For the case when $(p(S) - q(S)) \geq 0$ it is obvious that $d_{TV} = max_{S\subseteq[n]}|p(S) - q(S)| = max_{S\subseteq[n]}(p(S) - q(S))$. However, when $(p(S) - q(S)) < 0$, we argue as follows:

$$\begin{aligned} max_{S\subseteq[n]}|p(S) - q(S)| &= max_{S\subseteq[n]}q(S) - p(S) \xrightarrow{p(S)+p(\hat{S})=1 \text{ and } q(S)+q(\hat{S})=1} \\ &= max_{S\subseteq[n]}q(S) - p(S) + (p(S) + p(\hat{S})) - (q(S) + q(\hat{S})) \Longrightarrow \\ &= max_{S\subseteq[n]}p(\hat{S}) - q(\hat{S}) \xrightarrow{\hat{S} \text{ draws samples from the same set } [n]} \\ &= max_{S\subseteq[n]}p(S) - q(S) \end{aligned}$$

Therefore, we can conclude that both branches gives $d_{TV} = max_{S\subseteq[n]}|p(S) - q(S)| = max_{S\subseteq[n]}p(S) - q(S)$.

b)

First, I define as $\hat{S}$ the complimentary subset of [n] such that $p(S) + p(\hat{S}) = 1$ and $q(S) + q(\hat{S}) = 1$. Given from the previous sub question that $d_{TV} = max_{S\subseteq[n]}(p(S) - q(S))$ we can transform it to answer the current sub question as follows:

$$\begin{aligned} d_{TV} &= max_{S\subseteq[n]}p(S) - q(S) \xrightarrow{p(S)+p(\hat{S})=1 \text{ and } q(S)+q(\hat{S})=1} \\ &= max_{S\subseteq[n]}(p(S) - q(S) + q(S) + q(\hat{S}) - (p(S) + p(\hat{S})) \Longrightarrow \\ &= max_{S\subseteq[n]}q(\hat{S}) - p(\hat{S}) \xrightarrow{\hat{S} \text{ draws samples from the same set } [n]} \\ &= max_{S\subseteq[n]}q(S) - p(S) \Longrightarrow \\ &= -max_{\hat{S}\subseteq[n]}p(\hat{S}) - q(\hat{S}) \xrightarrow{*} \\ &= -min_{S\subseteq[n]}p(S) - q(S) \end{aligned}$$

(*) At the last step, Maximizing the complimentary subset $\hat{S}$ is equal to minimizing the initial subset S.

c)

To prove it is by defining two types of sets, S and $\hat{S}$. In particular, S type of set includes all the events in which $||p - q|| = max_{S\subseteq[n]}p(S) - q(S)$ while for $\hat{S}$ type of set we include all the events in which $||p - q|| = max_{S\subseteq[n]}q(\hat{S}) - p(\hat{S})$.

In order to answer, this sub question I draw in the figure 1 above, without loss of generality two discrete distributions p and q. The figure presents the euclidean distances between those two distributions. At

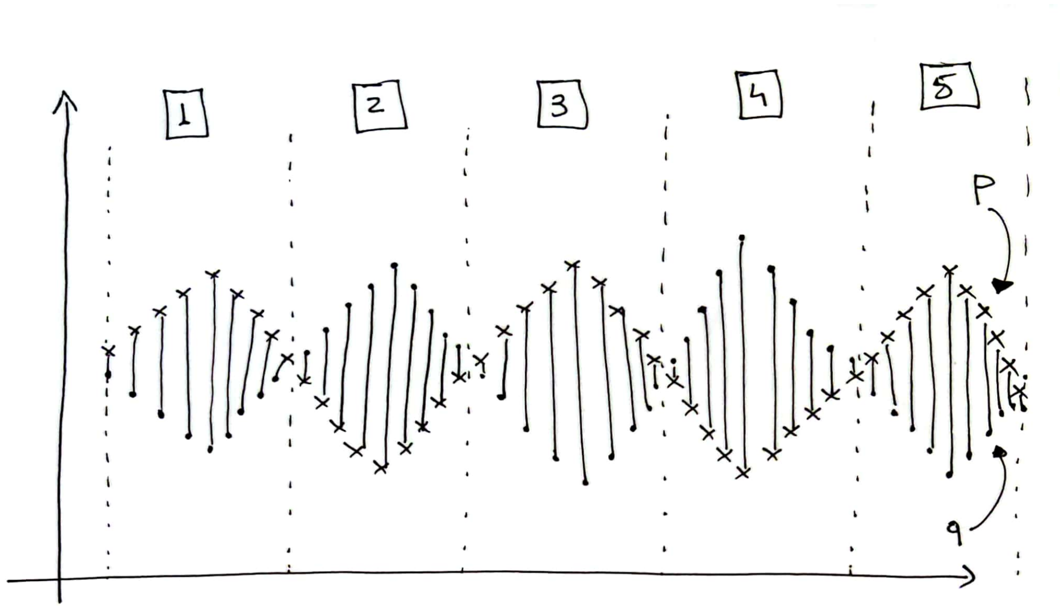

Figure 1: Question 1c Example with distributions p and q

slots 1, 3 and 5 the respective euclidean distances are described from the sub question a) since $||p - q|| = max_{S\subseteq[n]}(p(S) - q(S)) = d_{TV}$. At slots 2 and 4, the respective euclidean distances are described from the sub question a) since $||p - q|| = max_{\hat{S}\subseteq[n]}(q(\hat{S}) - p(\hat{S})) = -min_{S\subseteq[n]}(p(S) - q(S)) = d_{TV}$. In total, for all slots or in other words, for every point of the respected distributions we have that the euclidean distances between all the points are described as follows:

$$\begin{aligned} ||p - q|| &= max_{S\subseteq[n]}(p(S) - q(S)) + max_{\hat{S}\subseteq[n]}(q(\hat{S}) - p(\hat{S})) \\ &= max_{S\subseteq[n]}(p(S) - q(S)) - min_{S\subseteq[n]}(p(S) - q(S)) \xrightarrow{\text{sub question a) and b)}} \\ &= d_{TV} + d_{TV} \Longrightarrow \\ d_{TV} &= \frac{||p - q||}{2} \end{aligned}$$

2

The pseudo code is provided in python language and the data structure that is utilized is a set.

```python
def matching_func(edges):
    # Only a set data structure to achive O(n) space
    large_matching = set()
    # Check all edges if both endpoints are not included in the final output matching
    # If an edge is not then add it to output matching.
    for edge in edges:
        src = edge[0]
        dst = edge[1]
        if src not in large_matching and dst not in large_matching:
            large_matching.add(edge)
    return large_matching
```

It is easily observed that after passing all the edges the output matching will be a maximal matching, since there will be not any more edge that can be added with at least one endpoint that had not been already included in the output set. The last step to complete the answer of this question is to prove that the matching of my algorithm is at least half size compared to the maximum matching of any given graph.

First, given a graph $G(V,E)$, I define the maximal matching as M and the maximum matching as $\hat{M}$. Moreover, I define the number of edges for each matching as $E(M)$ and $E(\hat{M})$ respectively, while the number of nodes covered for each matching is defined as $N(M)$ and $N(\hat{M})$. What we want to prove is $E(M) \geq \dfrac{E(\hat{M})}{2}$.

Suppose that $E(M) = a$ and $E(\hat{M}) = b$ with $a, b \geq 0$, then the number of nodes covered for each matching is $N(M) = 2 \cdot a$ and $N(\hat{M}) = 2 \cdot b$. Moreover, we know that for each edge in E, a maximal matching will cover at least one node. Furthermore, we observe that for each edge in $\hat{M}$ at least 1 node of M is covered. Given the fact that $E(\hat{M}) = b$ then at least b nodes are included both from M and $\hat{M}$.Therefore, it is true that:

$$\begin{aligned} N(M) = 2 \cdot a \geq b \Longrightarrow \\ a \geq \frac{b}{2} \xrightarrow{E(M)=a \text{ and } E(\hat{M})=b} E(M) \geq \frac{E(\hat{M})}{2} \end{aligned}$$

5

a)

I define as X_i, the Bernoulli variable in which the sample is in $\{s_1, ..., s_a\}$ out of $C \cdot \epsilon^{-2}$ elements. Moreover, we define random variable X as the sum of all the X_i's. Based on the definitions we want to show that $Pr(X \geq \frac{1}{2} \cdot (1 + \frac{\epsilon}{2}) \cdot C \cdot \epsilon^{-2}) \geq 0.99$.
Calculating the expected values, we have:

$$\begin{aligned} E[X_i] = Pr(X_i = 1) = \frac{a}{n} \Longrightarrow \\ E[X] = E[X_1] + ... + E[X_n] = C \cdot \epsilon^{-2} \cdot \frac{a}{n} = \frac{C}{2} \cdot \epsilon^{-2} \cdot (1 + \epsilon) \end{aligned}$$

Calculating the error is the following:

$$\begin{aligned} Pr(error) = 1 - Pr(X \geq \frac{1}{2} \cdot (1 + \frac{\epsilon}{2})) = Pr(X < \frac{1}{2} \cdot (1 + \frac{\epsilon}{2})) \xrightarrow{\text{Chernoff bound}} \\ = Pr(X < (1 - \delta) \cdot E[X]) \leq e^{-\delta^2 \cdot \frac{E[X]}{2}} = 0.01 \end{aligned}$$

For δ:

$$\begin{aligned} (1 - \delta) \cdot E[X] = \frac{1}{2} \cdot (1 + \frac{\epsilon}{2}) \cdot C \cdot \epsilon^{-2} \xrightarrow{E[X] = \frac{C}{2} \cdot \epsilon^{-2} \cdot (1+\epsilon)} \\ (1 - \delta) \cdot \frac{C}{2} \cdot \epsilon^{-2} \cdot (1 + \epsilon) = \frac{1}{2} \cdot (1 + \frac{\epsilon}{2}) \cdot C \cdot \epsilon^{-2} \\ \delta = \frac{\epsilon}{2} \cdot \frac{1}{(1 + \epsilon)} \end{aligned}$$

Combining $\delta = \frac{\epsilon}{2} \cdot \frac{1}{(1+\epsilon)}$ with $Pr(error)$ we have:

$$\begin{aligned}
Pr(error) = Pr(X < (1-\delta) \cdot E[X]) \leq e^{-\delta^2 \cdot \frac{E[X]}{2}} = 0.01 \implies \\
\leq e^{-\frac{\epsilon}{2} \cdot (\frac{1}{(1+\epsilon)})^2 \cdot \frac{E[X]}{2}} \leq 0.01 \implies \\
\leq e^{-\frac{1}{16} \cdot \frac{1}{(1+\epsilon)} \cdot C} \leq 0.01 \implies \\
C \geq ln(100) \cdot (1+\epsilon) \cdot 16 \xrightarrow{\epsilon_{max}=0.1} \\
C_{max} \gtrapprox 81.04
\end{aligned}$$

Given the fact that the C is sufficient large that means that the statement for $Pr(error) \leq 0.01$ holds.

b)

Following the same pattern with sub question a), I define as X_i, the Bernoulli variable in which the sample is in $\{s_1, ..., s_{b-1}\}$ out of $C \cdot \epsilon^{-2}$ elements. Moreover, we define random variable X as the sum of all the X_i's. Based on the definitions we want to show that $Pr(X \leq (1 - \frac{1}{2} \cdot (1 + \frac{\epsilon}{2})) \cdot C \cdot \epsilon^{-2}) \geq 0.99$. In other words, I use the same idea as sub question a but I define the problem based on the complimentary part of $C \cdot \epsilon^{-2}$ elements.

Calculating the expected values, we have:

$$\begin{aligned}
E[X_i] = Pr(X_i = 1) = \frac{b-1}{n} \implies \\
E[X] = E[X_1] + ... + E[X_n] = C \cdot \epsilon^{-2} \cdot \frac{b-1}{n} = C \cdot \epsilon^{-2} \cdot \frac{\frac{n}{2} \cdot (1-\epsilon) - 1}{n}
\end{aligned}$$

Calculating the error is the following:

$$\begin{aligned}
Pr(error) = 1 - Pr(X \leq (1 - \frac{1}{2} \cdot (1 + \frac{\epsilon}{2})) \cdot C \cdot \epsilon^{-2}) = Pr(X > (1 - \frac{1}{2} \cdot (1 + \frac{\epsilon}{2})) \cdot C \cdot \epsilon^{-2}) \xrightarrow{\text{Markov inequality}} \\
\leq \frac{E[X]}{(1 - \frac{1}{2} \cdot (1 + \frac{\epsilon}{2})) \cdot C \cdot \epsilon^{-2}} \xrightarrow{E[X] = C \cdot \epsilon^{-2} \cdot \frac{\frac{n}{2} \cdot (1-\epsilon) - 1}{n}} \\
\leq \frac{C \cdot \epsilon^{-2} \cdot \frac{\frac{n}{2} \cdot (1-\epsilon) - 1}{n}}{(1 - \frac{1}{2} \cdot (1 + \frac{\epsilon}{2})) \cdot C \cdot \epsilon^{-2}} \implies \\
\leq \frac{n \cdot (1-\epsilon) - 2}{n \cdot (1 - \frac{\epsilon}{2})} \xrightarrow{n \geq 1000 \cdot \epsilon^{-2}, \epsilon = 0.1} \\
\leq 0.01
\end{aligned}$$

Given the fact that the statement for $Pr(error) \leq 0.01$ holds it is true that $Pr(X \leq (1 - \frac{1}{2} \cdot (1 + \frac{\epsilon}{2})) \cdot C \cdot \epsilon^{-2}) \geq 0.99$.

c)

Given that sub question a and b holds, the figure below holds. In particular, the previous two sub questions define if both holds then by definition of $\frac{\epsilon}{2}$ approximate median of the sample, the median will be inside this

Emmanouil Kritharakis

range. If we define as p the probability that sub question a) holds and q the probability that sub question b) holds, given the fact that those events are independent we state the join probability that defines the requested probability of current sub questions as follows:

$$
\begin{aligned}
Pr(\epsilon\text{-approximate median of S}) &= Pr(\text{sub question a) holds, sub question b) holds}) \xrightarrow{\text{a) \& b) are independent events}} \\
&= Pr(\text{sub question a holds}) \cdot Pr(\text{sub question b holds}) \\
&= p \cdot q = (\frac{99}{100})^2 \approx \frac{98}{100}.
\end{aligned}
$$

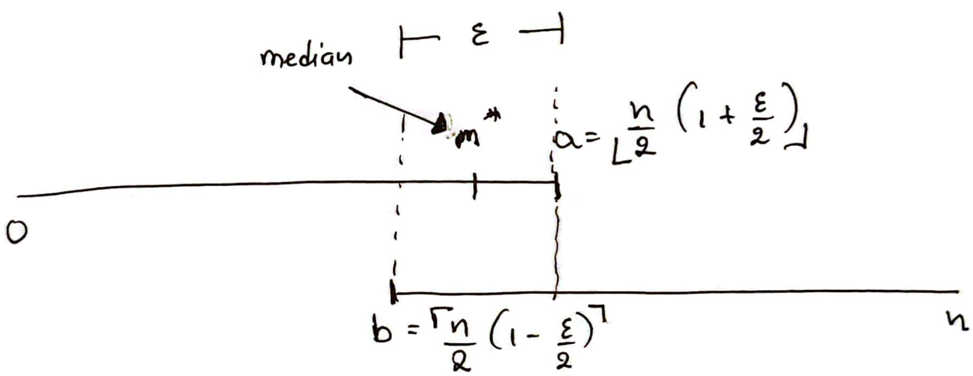

Figure 2: $\frac{\epsilon}{2}$ approximate median based on sub questions a and b.

6

I spent 8-10 days in total to answer the questions and write my solutions. The homework was neither easy nor hard. It had both easy questions and hard questions to answer.