Natural Language Understanding with Privacy-Preserving BERT

1. 什么是d-χ隐私（一种局部差分隐私的变体），与LDP（本地差分隐私）的区别

   d-χ隐私：根据向量之间的相似性来决定噪声大小，系统可以根据购买记录之间的相似性调整添加的噪音量。例如，如果两个用户购买了非常类似的物品组合，他们的记录在添加噪音后仍然会保持相似，因为它们在**原始数据中就很接近**。这种情况下，**噪音量会相对较小**。相反，如果另外两个用户的**购买记录差异很大**，那么在添加噪音后，这些记录在统计上会显得更加不同，因为原始数据中的差异已经较大。这样一来，d-χ隐私允许在保护用户隐私的同时，为具有相似购买行为的用户生成**更准确**的推荐。

   大创有关的想法：能否换成d-χ隐私？也许mae不会上升太多

   LDP：无论购买记录是否相似，都会应用**相同程度**的“噪音”来保护隐私。

2. 如何实现d-χ隐私？

```python
import numpy as np

def add_d_chi_privacy(user_vector, epsilon, distance_metric, sensitivity):
    """
    Add d-chi privacy noise to a user vector.

    Parameters:
    user_vector (np.array): Original user vector.
    epsilon (float): Privacy budget.
    distance_metric (function): A function to compute the distance between
vectors.
    sensitivity (float): The sensitivity of the data.

    Returns:
    np.array: User vector after adding noise.
    """

    # Calculate the scale of the Laplace noise based on epsilon and
sensitivity
    scale = sensitivity / epsilon

    # Calculate the distance of the user vector from a reference point
(e.g., origin)
    distance = distance_metric(user_vector, np.zeros(user_vector.shape))

    # Adjust the scale based on the distance (d-chi privacy)
    adjusted_scale = scale * distance

    # Generate the Laplace noise
    noise = np.random.laplace(0, adjusted_scale, size=user_vector.shape)

    # Add noise to the user vector
    noisy_vector = user_vector + noise
```

```
    return noisy_vector

# Example usage
user_vector = np.array([1.5, 2.5, 3.5])  # Example user vector
epsilon = 1.0  # Privacy budget
sensitivity = 1.0  # Sensitivity of the data

# Define a simple Euclidean distance metric
def euclidean_distance(vec1, vec2):
    return np.linalg.norm(vec1 - vec2)

# Add d-chi privacy noise to the user vector
noisy_vector = add_d_chi_privacy(user_vector, epsilon, euclidean_distance,
sensitivity)
noisy_vector
```

3. 在以上代码中为什么以全零向量作为基准?

   为了量化用户向量与一个标准或中性基准点的距离。全零向量通常被选作参考点，因为它在许多上下文中被视为一种"空"或"中性"状态，没有任何特定的属性或倾向。在应用 d-χ隐私时，计算用户向量与原点的距离可以帮助确定向量与"无信息状态"的差异程度。这种差异程度可以用来调整根据隐私需求添加到向量中的噪音量。然而，具体选择哪个参考点可能取决于应用的上下文和数据的性质。