基于 LCS 算法计算网络流量的讨论

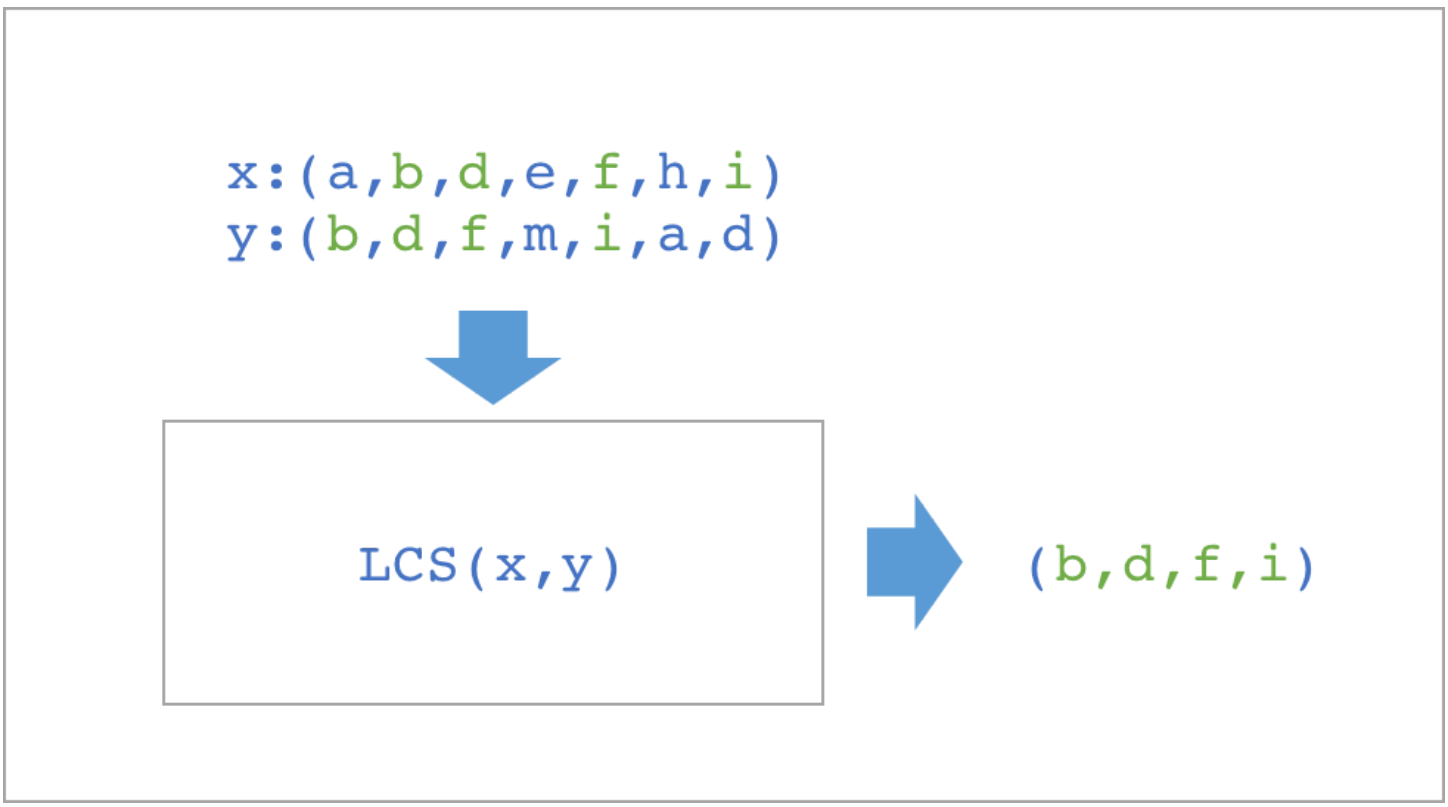
概要

假设 APP 开启时会产生特定的域名序列（或可化约的域名组合），我们可以提取这类域名序列作为模版，通过 LCS（最长公共子序列） 算法匹配每个设备的模版匹配程度，并汇总计算 APP 访问流量。

LCS 最大公共子序列

$S_1 : (u_1, u_2, u_3, ..., u_n)$
$S_2 : (v_1, v_2, v_3, ..., v_n)$

$LCS(S_1, S_2) = longest\ common\ subsequence$



模版库匹配

符号定义

$S : (s_1, s_2, ..., s_n)$ APP 开启时的域名序列

s_{adj} 经调整后 APP 开始时的模版

$d_i:(d_{i1},d_{i2},...,d_{im})$ 设备i单日域名访问记录

$D:(d_1,d_2,...,d_j)$ 所有设备单日域名访问记录

$LCS(s_{adj},d)=sd$ 最大匹配序列

$LR=\frac{len(sd)}{len(s_{adj})}$ 匹配度

$time\_span(sd)$ 匹配序列时间跨度

网络流量计算任务流程

1. 生成 APP 匹配模版库
2. 就单一设备计算匹配长度和时间跨度
3. 汇总 UV

计算量评估

n：模版长度 m_j：设备j单日域名访问长度

$\sum_{j=1} n\,m_j$

UV 计算

```
uv=0
for d in D:
    sd=LCS(s,d)
    # 时间跨度小于 1 s,最大匹配度大于 80%(需验证)
    if time_span(sd) < 1s and LR(sd)>80%:
        uv+=1
```

LCS 算法（不完备）

理论上可能存在多条匹配结果，代码仅展示其一

```
def lcs(X, Y, m, n):
    L = [[0 for x in range(n+1)] for x in range(m+1)]

    for i in range(m+1):
        for j in range(n+1):
            if i == 0 or j == 0:
                L[i][j] = 0
            elif X[i-1] == Y[j-1]:
                L[i][j] = L[i-1][j-1] + 1
            else:
                L[i][j] = max(L[i-1][j], L[i][j-1])
    index = L[m][n]
    lcs = [""] * (index)
    i = m
    j = n
    while i > 0 and j > 0:
        if X[i-1] == Y[j-1]:
            lcs[index-1] = X[i-1]
            i-=1
            j-=1
            index-=1

        elif L[i-1][j] > L[i][j-1]:
            i-=1
        else:
            j-=1

    return lcs
```