Finding Diverse Strings and Longest Common Subsequences in a Graph

Luca Lombardo

Seminar for the course of Bioinformatics

Structure of the Presentation

Longest Common Subsequence (LCS)

Definition

Given a set of m strings $S = \{S_1, S_2, \ldots, S_m\}$, a **common subsequence** (CS) is a sequence that appears in all m strings. A **longest common subsequence** LCS is a common subsequence of maximum length. We denote the set of all LCSs of S as $LCS(S)$.

Goal

The goal is to find a diverse set of solutions to the LCS problem under the Hamming distance.

Definition

Given two strings $X, Y \in \Sigma^r$, the **Hamming distance** between X and Y, denoted with $d_H(X, Y)$, is the number of positions at which the corresponding symbols differ.

A simple example

Add here the example of the paper

Efficient methods for finding a diverse set of solutions

More formally, let's consider the following two diversity measures for a multiset $\mathcal{X} = \{X_1, X_2, \dots, X_K\} \subseteq \Sigma^r$ of solutions, allowing repetitions:

$$D^{\text{sum}}_{d_H}(\mathcal{X}) = \sum_{1 \le i < j \le K} d_H(X_i, X_j) \qquad \textsc{Max-Sum Diversity} \qquad (1)$$

$$D^{\text{min}}_{d_H}(\mathcal{X}) = \min_{i<j} d_H(X_i, X_j) \qquad \textsc{Max-Min Diversity} \qquad (2)$$

Notation

A subset $\mathcal{X} \subseteq \Sigma^r$ is Δ-diverse w.r.t. $D^{\tau}_{d_H}$ if $D^{\tau}_{d_H}(\mathcal{X}) \ge \Delta$ for some $\Delta \ge 0$.

Where for $\tau \in \{sum, min\}$, $D^{\tau}_{d_H}$ denotes one of the two diversity measures.

Two problems

Problem 1: Diverse LCSs with Diversity measure $D^{\tau}_{d_H}$

Input: A set $S = \{S_1, S_2, \ldots, S_m\}$ of $m \geq 2$ strings over Σ, an integer $K \geq 1$ and $\Delta \geq 0$.
Question: Is there some set $\mathcal{X} \subseteq LCS(S)$ such that $\mathcal{X} = K$ and $D^{\tau}_{d_H}(\mathcal{X}) \geq \Delta$?

Problem 2: Diverse String Set

Input: $K, r, \Delta \in \mathbb{Z}$ and a Σ-DAG G for a set $L(G) \subseteq \Sigma^r$ of strings.
Question: Decide if there exists some subset $\mathcal{X} \subseteq L(G)$ such that $|\mathcal{X}| = K$ and $D^{\tau}_{d_H}(\mathcal{X}) \geq \Delta$.

Σ-Labeled Directed Acyclic Graphs (Σ-DAGs)

Definitions

- **Alphabet (Σ)**: Set of symbols.
- **String Set (Language)**: $L = \{X_1, X_2, \ldots, X_n\} \subseteq \Sigma^*$, with:
  - **Total Length**: $||L|| = \sum_{X \in L} |X|$
  - **Max Length**: $\mathrm{maxLen}(L) = \max_{X \in L} |X|$
- **r-String**: Any string X where $|X| = r$

Σ-DAG Structure

A graph $G = (V, E, s, t)$ with:

- V: vertices, E: labeled edges (v, c, w) with $c \in \Sigma$
- **Source** s and **Sink** t such that paths exist from s to all vertices
- **Size**: $\mathrm{size}(G)$, the number of its labeled edges

Σ-Labeled Directed Acyclic Graphs (Σ-DAGs)

Any path $P = (e_1, e_2, \ldots, e_n)$ of outgoing edges *spells out* a string str(P) $= c_1 c_2 \ldots c_n \in \Sigma^n$ where c_i is the label of edge e_i.

Language Representation

A Σ-DAG represents $L(G) \subset \Sigma^*$: all strings spelled from paths $s \to t$.
Equivalent to an NFA over Σ with initial s, final t, and no ϵ-edges.

Remarks

- For any set L of strings, a Σ-DAG G exists s.t. $L(G) = L$ and size$(G) \leq ||L||$. Construction time: $O(||L|| \log |\Sigma|)$.
- If G represents a set L of r-strings ($L \subseteq \Sigma^r, r \geq 0$), all paths $s \to v$ have the same length $d \leq r$.

Σ-DAGs for LCSs and Diverse String Sets

Lemma (Σ-DAG for LCSs)

For any constant $m \geq 1$ and set $S = \{S_1, \ldots, S_m\} \subseteq \Sigma^*$ of m strings, there exists a Σ-DAG G of polynomial size in $\ell := \text{maxlen}(S)$ such that $L(G) = \text{LCS}(S)$ and can be computed in polynomial time l

Consequence of the Lemma

- **If** Max-Min (or Max-Sum) Diverse String Set solvable in $f(M, K, r, \Delta)$,
- **Then** Max-Min (or Max-Sum) Diverse LCSs on $S \subseteq \Sigma^r$ solvable in $O(|\Sigma| \cdot \ell^m + f(\ell^m, K, r, \Delta))$ time.

where $\ell = \text{maxlen}(S)$

Exact Algorithms for Bounded Number of Diverse Strings

Algorithm for Max-Min Diverse String Set problem

Dynamic Programming Approach

Pattern of Path Tuple: For each $d \leq r$ and K-tuple of length-d paths $P = (P_1, \dots, P_K)$, define pattern $\text{Pattern}(P) = (\mathbf{w}, \mathbf{Z})$, where:

- $\mathbf{w} = (w_1, \dots, w_K)$: K-tuple of vertices representing endpoints of P_i paths.
- $\mathbf{Z} = (Z_{i,j})$: Upper triangular matrix of Hamming distances, where $Z_{i,j} = \min\{\Delta, d_H(\text{str}(P_i), \text{str}(P_j))\}$.

DP Table of Weights

$$\text{Weights} : V^K \times (\Delta \cup \{0\})^{K \times K} \rightarrow \{0, 1\} \quad (3)$$

Boolean matrix where $\text{Weights}(\mathbf{w}, \mathbf{Z}) = 1$ iff $(\mathbf{w}, \mathbf{Z})$ matches pattern $\text{Pattern}(P)$ for some K-tuple of paths of length d from s to $\mathbf{w}$.

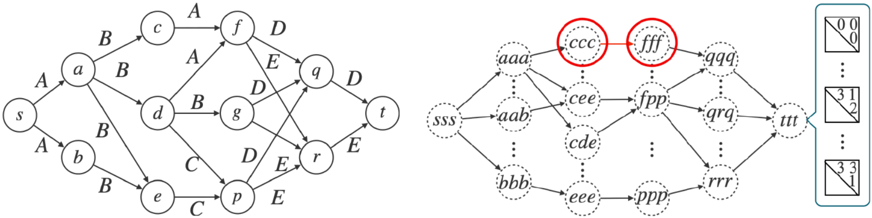
Algorithm for Max-Min Diverse String Set problem

Algorithm 1

Set $\texttt{Weights}(\mathbf{s}, Z) = 0$ for all $Z \in (\Delta \cup \{0\})^{K \times K}$ and $\texttt{Weights}(\mathbf{s}, \mathbf{0}) \leftarrow 1$
for $d \leftarrow 1, \ldots, r$ **do**
    for $\mathbf{v} \leftarrow (v_1, \ldots v_K) \in (V_d)^K$ **do**
        for $(v_1, c_1, w_1) \in E^+(v_1), \ldots, (v_K, c_K, w_K) \in E^+(v_K)$ **do**
            Set $\mathbf{w} = (w_1, \ldots, w_K)$
            for $U \in (\Delta \cup \{0\})^{K \times K}$ such that $\texttt{Weights}(\mathbf{v}, U) = 1$ **do**
                Set $Z = (Z_{i,j})_{i<j}$ with $Z_{i,j} \leftarrow \min\{\Delta, U_{i,j} + \mathbb{1}\{c_i \neq c_j\}\} \quad \forall\, i, j \in K$
                Set $\texttt{Weights}(\mathbf{w}, Z) \leftarrow 1$ ▷ Update
            end for
        end for
    end for
end for
Answer `Yes` if $\texttt{Weights}(\mathbf{t}, Z) = 1$ and $D^{\min}_{d_H}(Z) \geq \Delta$ for some Z, else `No`

For any $K \geq 1$ and $\Delta \geq 0$, it solves the Max-Min Diverse String Set problem in $O(\Delta^{K^2} K^2 M^K (\log|V| + \log \Delta))$ time and space when an input string set L is represented by a Σ-DAG G with size$(G) = M$

Example Run of the Algorithm



(a) An input Σ-DAG G_1 for $LCS(X_1, Y_1)$ **(b)** Example run of Algorithm 1 on G_1

Figure: (a) An input Σ-DAG G_1 over $\Sigma = \{A, B, C, D, E\}$ for the set of all longest common subsequences of two strings $X_1 = \text{ABABCDDEE}$ and $Y_1 = \text{ABCBAEEDD}$ and (b) an example run of Algorithm 1 based on dynamic programming with $K = 3$ on an input G_1

Algorithm for MAX-SUM DIVERSE STRING SET problem

Modify the Max-Sum Diverse String Set problem

Instead of the entire $K \times K$ weight matrix Z, only the sum $z = \sum_{i<j} d_H(\text{str}(P_i), \text{str}(P_j))$ is needed for computing Max-Sum diversity.

New DP Table Weights

For $\mathbf{w} = (w_1, \ldots, w_K)$ of depth $0 \leq d \leq r$ and integer $0 \leq z \leq rK$, define:

$$\texttt{Weights}(\mathbf{w}, z) = 1$$

if and only if there exists a K-tuple of length-d prefix paths $(P_1, \ldots, P_K)$ from s to $w_1, \ldots, w_K$ with sum of pairwise Hamming distances z.

Algorithm for Max-Sum Diverse String Set problem

Algorithm 2

```
Set Weights(s, Z) = 0 for all Z ∈ (Δ ∪ {0})^{K×K} and Weights(s, 0) ← 1
for d ← 1, ..., r do
    for v ← (v_1, ... v_K) ∈ (V_d)^K do
        for (v_1, c_1, w_1) ∈ E^+(v_1), ..., (v_K, c_K, w_K) ∈ E^+(v_K) do
            Set w = (w_1, ..., w_K)
            for u ← (0, ..., rK) such that Weights(v, U) = 1 do
                Set Z = (Z_{i,j})_{i<j} with Z_{i,j} ← min{Δ, u + Σ_{i<j} 𝟙{c_i ≠ c_j}}
                Set Weights(w, Z) ← 1                                   ▷ Update
            end for
        end for
    end for
end for
Answer Yes if Weights(t, Z) = 1 and D^{min}_{d_H}(Z) ≥ Δ for some Z, else No
```

For any $K \geq 1$, 15 solves the Max-Sum Diverse String Set under Hamming Distance in $O(\Delta K^2 M^K(\log|V| + \log\Delta))$ time and space, where M is the size of the input Σ-DAG G

Approximation Algorithms for Unbounded Number of Diverse Strings

Max-Min Diverse String Set problem

Use a local search algorithm for computing approximate solutions $\mathcal{X} \subseteq \mathcal{L}$ with $|\mathcal{X}| = K$ on a finite metric space $(\mathcal{L}, d)$, where $d : \mathcal{L} \times \mathcal{L} \to \mathbb{R}_{\geq 0}$.

Algorithm 3 $\texttt{LocalSearch}(\mathcal{L}, K, d)$

$\mathcal{X} \leftarrow$ arbitrary set of K solutions in $\mathcal{L}$
for $i \leftarrow 1, \ldots, \lceil \frac{K(K-1)}{K+1} \log \frac{(K+2)(K-1)^2}{4} \rceil$ **do**
  for $X \in \mathcal{X}$ s.t $\mathcal{L} \setminus \{X\} \neq \emptyset$ **do**
    $Y \leftarrow \operatorname{argmax}_{Y \in \mathcal{L} \setminus \{X\}} \sum_{X' \in \mathcal{X} \setminus \{X\}} d(X', Y)$
    $\mathcal{X} \leftarrow \mathcal{X} \setminus \{X\} \cup \{Y\}$
  end for
end for
return $\mathcal{X}$

Theorem

When the distance d is a semi-metric of negative type over $\mathcal{X}$, then LocalSearch *has improved approximation ratio $(1 - \frac{2}{K})$ for any $K \geq 2$. The Hamming distance d_H over the set of r-strings is a semi-metric of negative type.*

Max-Sum Farthest r-String problem

How do we solve efficiently the subproblem at line 4 of the LocalSearch algorithm?

Lemma (Max-Sum Farthest r-String)

For any $K \geq 1$ and $\Delta \geq 0$, LocalSearch computes the farthest r-string $Y \in L(G)$ that maximizes $D^{sum}_{d_H}(Y \cup \mathcal{X})$ over all r-strings in $L(G)$ in $O(K\Delta M)$ time and space, where M is the size of the input Σ-DAG G.

Theorem (Polynomial Time Approximation Scheme for unbounded K)

When K i part of an input, Max-Sum Diverse String Set problem on a Σ-DAG admits a PTAS

The corresponding result for the Max-Sum Diverse LCSs follows immediately.

Fixed-Parameter Tractable (FPT) Algorithms for Bounded Number and Length of Diverse Strings

FPT Algorithms for Max-Min and Max-Sum Diverse String Set problems

Definition

A problem parametrized with κ is said to be *fixed-parameter tractable* (FPT) if there exists an algorithm for the problem running on an input x in time $f(\kappa(x) \cdot |x|)$, where f is a computable function and $c > 0$ is a constant.

Theorem

When r, K, Δ are part of the input, the Max-Min Diverse String Set *problem on a Σ-DAG admits an FPT algorithm.*

Theorem

When r, K, Δ are part of the input, the Max-Sum Diverse String Set *problem on a Σ-graphs for r-strings admits an FPT algorithm.*

Conclusions

- When K is bounded, both MAX-MIN and MAX-SUM of both DIVERSE STRING SET and DIVERSE LCSS are solvable in polynomial time using dynamic programming.
- When K is part of the input, the MAX-SUM version of the problems admits a PTAS by local search showing that the Hamming distance is a semi-metric of negative type.