Project One

CSCI 8610 Topics in Theoretical Computer Science
(Fall 2014)

Computer Science, University of Georgia

This project explores the capability of stochastic regular grammar (SRG) to reveal hidden patterns of bimolecular sequences. The project contains two components of assignments: (1) development of a VITERBI Algorithm-like decoding algorithm for the general case of SRG; and (2) Investigation of its effectiveness in applications.

1 Decoding Algorithm for SRG

Your decoding algorithm accepts as input a stochastic regular grammar G and a terminal string x, and computes the maximum probability for the grammar G to derive the string x. The algorithm should utilize a dynamic programming strategy to compute the maximum probability. To facilitate the dynamic programming algorithm design, you may define function $q(X, i)$ to be the maximum probability for variable X to derive the prefix of x up to the first i terminal symbols. Then you would need to derive recurrences for $q(X, i)$, including base case(s), to be the formulae for the bottom-up table filling task required by the dynamic programming.

In principle, your VITERBI algorithm should work for all stochastic regular grammars. However, you are allowed to assume the "normal form" of regular grammars in which rules are either $X \to aY$ (or $X \to Ya$ if you prefer), and $X \to a$, where X and Y represent variables and a represents a terminal in the alphabet Σ. It is easy to see that regular rules like $X \to abcY$ can be converted to rules of the "normal form". However, it is not directly clear how to remove *unit rule* like $X \to Y$. For how to resolving this issue, see the **Requirements** section.

To facilitate your programming, you may assume a meta language (notations), with which the input SRG is written. For example, to permit "words"

or "phrases" for variables in the grammar, you may use meta symbols "<" and ">" to delimit a word or a phrase, such as `<my variable 1>`.

Other than the VITERBI algorithm, you may also implement the algorithm FORWARD that computes the total probability for the SRG to produce the given string x.

2 Applications

You would like to show that algorithms like VITERBI and FORWARD with SRGs are effective for solving some problems. SRGs, much like HMMs, are suitable for profiling biological sequence patterns. For example, you may define an SRG to generate a set of strings of some average length whose contents are mostly conserved with some limited variations (due to evolution, for example). Then such a profile can be used to identify additional members for the profile through probability computation with VITERBI or FORWARD algorithm. Typically, given a string x, you can compute the maximum or total probability for any given string to match the profile.

You may also design an SRG to discriminate sequence data of certain property from those without the property. For example, your SRG may contain rules for generating sequences of a high GC content and and sequences of normal GC content contents. With the SRG model and its decoding algorithm, you can develop additional subroutines to scan a long genome sequence to identify segments of genomes that are of the high GC content.

3 Requirements

The homework requires you to understand how to use SRGs for modeling string sets of interest and to develop related dynamic programming algorithms (for decoding, for example). The minimum requirements are:

1. Design the VIBERTI algorithm for decoding of input strings with respect to a predefined SRG;
2. Argue that the unit rule $X \to Y$ can be removed from any regular grammar so that it can be equivalently converted into one with rules of form $X \to aY$ (or $X \to Ya$ if you prefer) and of form $X \to a$ only;
3. Implement the designed VITERBI algorithm correctly;
4. Apply your work to evaluate the effectiveness of the methodology.