Dynamic Programming-based Search Algorithms in NLP

Liang Huang, Google Research

Dynamic Programming (DP) is an important class of algorithms widely used in many areas of speech and language processing. It provides efficient solutions to seemingly intractable inference over exponentially-large spaces by sharing overlapping subproblems. Well-known examples of DP in our field include Viterbi and Forward-Backward Algorithms for finite-state models, CKY and Earley Algorithms for context-free parsing, and A* Algorithm for both. These algorithms are widely used to solve problems ranging from sequence labeling to word alignment to machine translation decoding.

With this overwhelming popularity, this tutorial aims to provide a better understanding of DP from both theoretical and practical perspectives. In the theory part, we try to unify various DP algorithms under a generic algebraic framework, where the above mentioned examples are merely special cases, and we can easily analyze their correctness and complexities. However, exact DP algorithms are often infeasible in practice due to time and space constraints. So in the practice part, we will survey several widely used tricks to reduce the size of the search space, including beam search, histogram pruning, coarse-to-fine search, and cube pruning. We will discuss these methods within the context of state-of-the-art large-scale NLP systems.

1 Outline

- Part A: Dynamic Programming on Lattices/Graphs under Semiring Framework
  - theory:
    - Motivations and Examples
    - Semirings
    - Viterbi Algorithm
    - Dijkstra and A* Algorithms
    - Comparison between Viterbi and Dijkstra/A* Algorithms
  - practice:
    - Beam Search and Histogram Pruning; E.g.: *Pharaoh* (phrase-based MT)

- Part B: Dynamic in Programming on Packed Forests under Hypergraph Framework
  - theory:
    - Hypergraphs; Examples in Parsing and Machine Translation
    - Generalized Viterbi Algorithm; CKY Parsing
    - Knuth and A* Algorithms

- practice:
    - A* in Practice: A* Parsing; beam A*; inadmissible heuristics
    - Coarse-to-Fine Search; Example: *Charniak and Berkeley parsers*
    - Cube Pruning; Example: *Hiero* (syntax-based decoding)

2 Target Audience

This tutorial is intended for researchers with *any* level of familiarity with dynamic programming. A basic understanding of the CKY Algorithm is recommended, but not required.

3 Brief Bio of the Presenter

Liang Huang is a Research Scientist at Google Research. He obtained his PhD from the University of Pennsylvania under Aravind Joshi and Kevin Knight (USC/ISI). His research interests include algorithms in parsing and translation, generic dynamic programming, and syntax-based machine translation. His work on "forest-based algorithms" received an Outstanding Paper Award at ACL 2008, as well as Best Paper Nominations at ACL 2007 and EMNLP 2008. He also loves teaching and was a recipient of the University Prize for Excellence in Teaching at Penn.