CS 7180 Ayush Singh

PGM Paper Report
CRF & M3N

Dual Decomposition

The paper starts by motivating us for a solution by stating the fact that without additional restrictions on the choice of potential functions, or which edges to include, finding the maximum probability assignment of a graphical model (also known as the MAP assignment) is NP-Hard. Various approximation algorithms that use either Linear Programming or Dual Decomposition also known as Lagrangian Relaxation converts the model into a constrained optimization problem solvable in polynomial time but are not really scalable. Author shows the application and performance gains in finding out the MAP with the help of two examples from protein structures and natural language processing. Thus, the need for a general purpose algorithm for MAP inference arises where Dual Decomposition comes into play as it divides model into tractable components, solving each one locally and passing messages iteratively to induce the components.

With dual decomposition, the original problem is broken up into smaller subproblems involving maximizations of each single node potential and each edge potential. Although these local maximizing assignments are easy to obtain, they are unlikely to agree with each other without modifying the potential functions. These modifications are provided by the Lagrange multipliers associated with agreement constraints. Paper describes two classes of algorithms, one based on a subgradient method and another based on block coordinate descent for minimizing the Lagrangian relaxations with each having their own pros and cons. Next they give an upper bound which even when not exact, can be extremely valuable when used together with a branch-and-bound procedure to find the MAP assignment. They show that the MAP inference method is agnostic to either Linear Programming or Dual Decomposition.

The paper shows promising techniques for speeding up the performance in practice, dual decomposition has best theoretical guarantees and can use very fast using MAP subroutines for solving large model components but it is still not the fastest and has a lot of tunable features e.g. larger lambda to improve quality, adjust step size, methods to construct candidate which increases the flexibility but introduces the complexity leading to a lot of time wasted in fine tuning the model.