A^* Search via Approximate Factoring

Aria Haghighi, John DeNero, Dan Klein
Computer Science Division
University of California Berkeley
{aria42, denero, klein}@cs.berkeley.edu

Abstract

We present a novel method for creating A^* estimates for structured search problems originally described in Haghighi, DeNero, & Klein (2007). In our approach, we project a complex model onto multiple simpler models for which exact inference is efficient. We use an optimization framework to estimate parameters for these projections in a way which bounds the true costs. Similar to Klein & Manning (2003), we then combine completion estimates from the simpler models to guide search in the original complex model. We apply our approach to bitext parsing and demonstrate its effectiveness.

Introduction

Inference tasks in natural language processing (NLP) often involve searching for an optimal output from a large set of structured outputs. Example output spaces include sentences (machine translation and automatic speech recognition), partitions (coreference analysis), and trees (syntactic parsing). For many complex models, selecting the highest scoring output for a given observation is slow or even intractable.

One general technique to increase efficiency while preserving optimality is A^* search (Hart, Nilsson, & Raphael 1968); however, successfully using A^* search is challenging in practice. The design of admissible (or nearly admissible) heuristics which are both effective (close to actual completion costs) and also efficient to compute is a difficult, open problem in most domains. As a result, most work on search has focused on non-optimal methods, such as beam search or pruning based on approximate models (Collins 1999), though in certain cases admissible heuristics are known (Zhang & Gildea 2006). For example, Klein & Manning (2003) show a class of projection-based A^* estimates, but their application is limited to models which have a very restrictive kind of score decomposition. In this work, we broaden their projection-based technique to give A^* estimates for models which do not factor in this restricted way.

Like Klein & Manning (2003), we focus on search problems where there are multiple projections or "views" of the output structure. We use general optimization techniques (Boyd & Vandenberghe 2005) to approximately factor a model over these projections. Solutions to the projected problems yield heuristics for the original model. This approach is flexible, providing either admissible or nearly admissible heuristics, depending on the details of the optimization problem solved. Furthermore, our approach allows a modeler explicit control over the trade-off between the tightness of the heuristic and its degree of inadmissibility (if any). We describe our technique in general and then apply it to the task of bitext parsing in NLP.

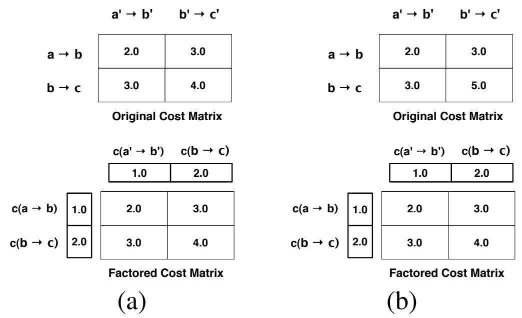

Figure 1: Example cost factoring: Each cell of the matrix is a local configuration composed of two projections (the row and column of the cell). In (a), the top matrix is an example cost matrix, which specifies the cost of each local configuration. The bottom matrix represents our factored estimates, where each entry is the sum of the configuration projections. For this example, the actual cost matrix can be decomposed exactly into two projections. In (b), the top cost matrix cannot be exactly decomposed along two dimensions. Our factored cost matrix has the property that each factored cost estimate is below the actual configuration cost. Although our factorization is no longer tight, it still can be used to produce an admissible heuristic.

General Approach

The search problem is to find a minimal cost path from the start state to a goal state, where the path cost is the sum of the costs of the edges in the path. When inferring an optimal structure under a probabilistic model, the cost of an edge is typically a negative log probability which depends only on some local configuration type. We will use a to refer to a local configuration and use $c(a)$ to refer to its cost. Because edge costs are sensitive only to local configurations, the cost of a path $\mathcal{P}$ is $\sum_{a \in \mathcal{P}} c(a)$. A* search requires a *heuristic function*, which is an estimate $h(s)$ of the *completion cost*, the cost of a best path from state s to a goal.

In this work, following Klein & Manning (2003), we consider problems with *projections* or "views," which define mappings to simpler state and configuration spaces. Formally, a projection π is a mapping from states to some coarser domain. A state projection induces projections of edges and of the entire graph $\pi(G)$.

We are particularly interested in search problems with multiple projections $\{\pi_1, \ldots, \pi_\ell\}$ where each projection, π_i, has the following properties: its state projections induce well-defined projections of the local configurations $\pi_i(a)$ used for scoring, *and* the projected search problem admits a simpler inference procedure than the original.

In defining projections, we have not yet dealt with the projected scoring function. Suppose that the cost of local configurations decomposes along projections as well,

$$c\,(a) = \sum_{i=1}^{\ell} c(\pi_i(a))\;, \forall a \in \mathcal{A} \qquad (1)$$

where $\mathcal{A}$ is the set of all local configurations.

A toy example of a score decomposition in the context of a Markov process over two-part states is shown in figure 1(a), where the costs of the joint transitions equal the sum of costs of their projections. Under the strong assumption of equation (1), Klein & Manning (2003) give an admissible A* bound. They note that the cost of a path decomposes as a sum of projected path costs. Hence, the following is an admissible A* heuristic for our problem,

$$h(s) = \sum_{i=1}^{\ell} h_i^*(\pi_i(s)) \qquad (2)$$

where $h_i^*(\pi_i(s))$ denote the optimal completion costs in the projected search graph $\pi_i(G)$. That is, the completion cost of a state bounds the sum of the completion costs in each projection.

In virtually all cases, however, configuration costs will not decompose over projections, nor would we expect them to. This independence assumption undermines the motivation for assuming a joint model over a complex structure with multiple projections. In the central contribution of this work, we exploit the projection structure of our search problem without making any assumption about cost decomposition.

Rather than assuming decomposition, we propose to find scores for the projected configurations which are *pointwise admissible*:

$$\sum_{i=1}^{\ell} \phi_i(\pi_i(a)) \leq c(a), \forall a \in \mathcal{A} \qquad (3)$$

Here, $\phi_i(\pi_i(a))$ represents the factored projection cost of $\pi_i(a)$, the π_i projection of configuration a. Given pointwise admissibility, we can again apply the heuristic recipe of equation (2). An example of factored projection costs are shown in figure 1(b), where no exact decomposition exists, but a pointwise admissible lower bound is easy to find.

Claim. *If a set of factored projection costs $\{\phi_1, \ldots, \phi_\ell\}$ satisfy pointwise admissibility, then the heuristic from (2) is an admissible A* heuristic.*

Proof. Assume $a_1, \ldots, a_k$ are configurations used to optimally reach the goal from state s. Then,

$$\begin{aligned} h^*(s) &= \sum_{j=1}^{k} c(a_j) \geq \sum_{j=1}^{k}\sum_{i=1}^{\ell} \phi_i(\pi_i(a_j)) \\ &= \sum_{i=1}^{\ell}\left(\sum_{j=1}^{k} \phi_i(\pi_i(a_j))\right) \geq \sum_{i=1}^{\ell} h_i^*(\pi_i(s)) = h(s) \end{aligned}$$

□

The first inequality follows from pointwise admissibility. The second inequality follows because each inner sum is a completion cost for projected problem π_i and therefore $h_i^*(\pi_i(s))$ lower bounds it.

Factored Projections for Non-Factored Costs

We can find factored costs $\phi_i(\pi_i(a))$ which are pointwise admissible by solving an optimization problem. We think of our unknown factored costs as a block vector $\phi = [\phi_1, .., \phi_\ell]$, where vector ϕ_i is composed of the factored costs, $\phi_i(\pi_i(a))$, for each configuration $a \in \mathcal{A}$. We can then find admissible factored costs by solving the following optimization problem,

$$\begin{aligned} &\underset{\gamma, \phi_i}{\text{minimize}} \;\; \|\gamma\| \qquad (4) \\ &\text{such that, } \gamma_a = c(a) - \sum_{i=1}^{\ell} \phi_i(\pi_i(a)), \forall a \in \mathcal{A} \\ &\gamma_a \geq 0, \forall a \in \mathcal{A} \end{aligned}$$

We can think of each γ_a as the amount by which the cost of configuration a exceeds the factored projection estimates (the pointwise A* gap). Requiring $\gamma_a \geq 0$ insures pointwise admissibility. Minimizing the norm of the γ_a variables encourages tighter bounds. In the case where we minimize the 1-norm or ∞-norm, the problem above reduces to a linear program, which can be solved efficiently for a large number of variables and constraints.

We could imagine many ways of deciding amongst the various admissible solutions. Viewing our procedure decision-theoretically, by minimizing the norm of the pointwise gaps we are effectively choosing a loss function which decomposes along configuration types and takes the form of the norm (i.e. linear or squared losses). A complete investigation of the possibilities is beyond the scope of this work, but it is worth pointing out that in the end we will care only about the gap on entire structures, not configurations, and individual configuration factored costs need not even be admissible for the overall heuristic to be admissible.

Notice that the number of constraints is $|\mathcal{A}|$, the number of possible local configurations. For many search problems, enumerating the possible configurations is not feasible, and therefore neither is solving an optimization problem with all of these constraints. We deal with this situation in applying our technique to lexicalized parsing models (Haghighi, DeNero, & Klein 2007).

Nearly Admissible Heuristics

Sometimes, we might be willing to trade search optimality for efficiency. In our approach, we can explicitly make this trade-off by designing an alternative optimization problem which allows for slack in the admissibility constraints. We solve the following soft version of problem (4):

$$\underset{\gamma,\phi_i}{\text{minimize}} \;\; \|\gamma^+\| + C\|\gamma^-\|_2 \tag{5}$$

$$\text{such that,}\;\; \gamma_a = c(a) - \sum_{i=1}^{\ell} \phi_i(\pi_i(a)), \forall a \in \mathcal{A}$$

where $\gamma^+ = \max\{0, \gamma\}$ and $\gamma^- = \max\{0, -\gamma\}$ represent the component-wise positive and negative elements of γ respectively. Each $\gamma_a^- > 0$ represents a configuration where our factored projection estimate exceeds the actual configuration cost. Since this situation may result in our heuristic becoming inadmissible if they are used in the projected completion costs, we more heavily penalize overestimating the cost by the constant C.

We note that we can bound our search error in this setting. Suppose $\gamma_{\max}^- = \max_{a\in\mathcal{A}} \gamma_a^-$ and that L^* is the length of the longest optimal solution for the original problem. This ϵ-admissible heuristics (Ghallab & Allard 1982) bounds our search error by $L^*\gamma_{\max}^-$.[1]

Bitext Parsing

In bitext parsing, we jointly infer a synchronous phrase structure tree over a sentence w_s and its translation w_t (Melamed, Satta, & Wellington 2004; Wu 1997). Recent advances in syntax-based machine translation have sparked renewed community interest in this task (Zhang & Gildea 2006; Galley *et al.* 2004).

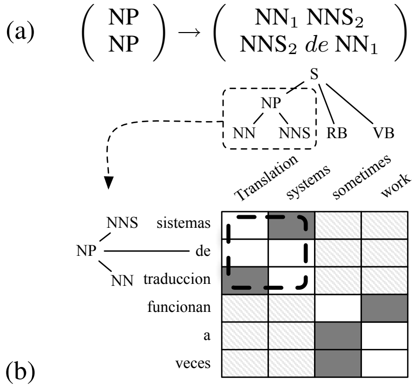

Figure 2: (a) A tree-to-tree transducer rule. (b) An example training sentence pair that yields rule (a).

Bitext parsing is a natural candidate task for our approximate factoring technique. A synchronous tree projects monolingual phrase structure trees onto each sentence, which can each be scored independently by a weighted context-free grammar (WCFG), providing our heuristic. However, the costs assigned by a weighted synchronous grammar (WSG) do not typically factor into independent monolingual WCFGs. We can, however, produce a useful surrogate: a pair of monolingual WCFGs with structures projected by a WSG and rule weights that, when combined, uniformly underestimate the weights of the WSG.

A Synchronous Grammar Formalism

We demonstrate our technique by parsing using the synchronous grammar formalism of tree-to-tree transducers (Knight & Graehl 2004). The weighted grammar rules model a generative process wherein an aligned pair of nonterminals jointly generate two ordered lists of children, which may contain either terminals or nonterminals. The non-terminals in each list must align one-to-one to the non-terminals in the other, while the terminals are placed freely on either side. Figure 2(a) shows an example rule. We can learn such a synchronous grammar by projecting English syntax onto a foreign language via a word-level alignment, as shown in figure 2(b). [2]

Parsing with a synchronous grammar via a dynamic program requires time $O(n^6)$ in the length of the sentence (Wu 1997). This high complexity makes exhaus-

[1] This bound may be very loose if L is large.

[2] Our grammar extraction protocol follows Galley *et al.* (2004). Given a sentence-aligned corpus we fix a word alignment using GIZA++ (Och & Ney 2000) and parses of the English sentences with a parser based on Klein & Manning (2003). We score these extracted synchronous rules using relative frequency estimates.

tive parsing infeasible for all but the shortest sentences. In contrast, monolingual CFG parsing is only $O(n^3)$.

A* Parsing

We now consider inferring the optimal parse under a WSG using A*. The states are rooted bispans, denoted:

$$X\,[i,j] :: Y\,[k,l]$$

States represent a joint parse over subspans $[i,j]$ of w_s and $[k,l]$ of w_t rooted by the nonterminals X and Y respectively. Given a WSG $\mathcal{G}$, the algorithm prioritizes a state s by the sum of its inside cost $\beta_{\mathcal{G}}(s)$ (the negative log of its inside probability) and its outside estimate $\hat{\alpha}(s)$, or completion cost.[3] We are guaranteed the optimal parse under $\mathcal{G}$ if our heuristic $\hat{\alpha}(s)$ is never greater than $\alpha_{\mathcal{G}}(s)$, the true outside cost of s. We achieve this bound by enforcing point-wise admissibility in the projections of $\mathcal{G}$ that yield $\hat{\alpha}(s)$, described below.

Projecting the Synchronous Grammar

$\mathcal{G}$ projects a pair of monolingual CFGs $\mathcal{G}_s$ and $\mathcal{G}_t$ given by splitting each synchronous rule

$$r = \left(\begin{array}{c} \mathrm{X} \\ \mathrm{Y} \end{array} \right) \rightarrow \left(\begin{array}{c} \alpha\,\beta \\ \gamma\,\delta \end{array} \right)$$

into two rules: $\pi_s(r) = X \rightarrow \alpha\beta$ and $\pi_t(r) = Y \rightarrow \gamma\delta$.

To learn pointwise admissible costs for the rules in $\mathcal{G}_s$ and $\mathcal{G}_t$, we solve the following optimization problem: [4]

$$\begin{aligned} &\underset{\gamma,\phi_s,\phi_t}{\text{minimize}} \;\; \|\gamma\|_1 \\ &\text{such that, } \gamma_r = c(r) - [\phi_s(\pi_s(r)) + \phi_t(\pi_t(r))] \\ &\qquad\qquad \text{for all synchronous rules } r \in \mathcal{G} \\ &\qquad\qquad \phi_s \geq 0, \phi_t \geq 0, \gamma \geq 0 \end{aligned}$$

We then compute an admissible heuristic by parsing with $\mathcal{G}_s$ and $\mathcal{G}_t$. Each synchronous state $s = X\,[i,j] :: Y\,[k,l]$ projects a pair of monolingual rooted spans. The heuristic we propose sums the independent outside costs of these spans in each monolingual projection:

$$\hat{\alpha}(s) = \alpha_s(X\,[i,j]) + \alpha_t(Y\,[k,l])$$

Experiments

We parsed 1200 English-Spanish sentences using a tree-to-tree transduction grammar learned from 40,000 sentence pairs. Figure 3(a) show that A* expands substantially fewer states while searching for the optimal parse with our heuristic. The *exhaustive* curve shows edge expansions using the null heuristic. The intermediate result, labeled *English only*, used only the English monolingual outside score as a heuristic. Similar results using only Spanish demonstrate that both projections contribute to parsing efficiency. All three curves in Figure 3 represent running times for finding the *optimal* parse.

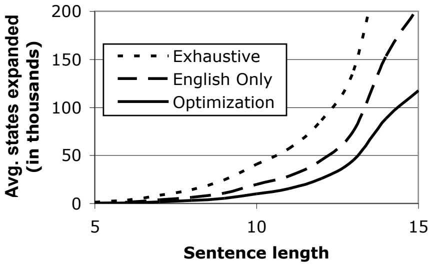

Figure 3: Parsing efficiency results show that the use of a suitable heuristic improves performance substantially, and both projections are required to maximize performance.

Conclusion

We have presented a technique for creating A* estimates for inference in complex models. Our technique can be used to generate provably admissible estimates when all search transitions can be enumerated, and an effective heuristic even for problems where all transitions cannot be efficiently enumerated. In the future we plan to investigate alternative objective functions and error-driven methods for learning heuristic bounds.

References

Boyd, S., and Vandenberghe, L. 2005. *Convex Optimization*. Cambridge University Press.

Collins, M. 1999. Head-driven statistical models for natural language parsing.

Galley, M.; Hopkins, M.; Knight, K.; and Marcu, D. 2004. What's in a translation rule? In *HLT-NAACL 2004: Main Proceedings*.

Ghallab, M., and Allard, D. G. 1982. A$^*_\epsilon$ - an efficient near admissible heuristic search algorithm. In *IJCAI*.

Haghighi, A.; DeNero, J.; and Klein, D. 2007. Approximate factoring for A* search. In *HLT-NAACL*. Association for Computational Linguistics.

Hart, P.; Nilsson, N.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. In *IEEE Transactions on Systems Science and Cybernetics*. IEEE.

Klein, D., and Manning, C. D. 2003. Factored A* search for models over sequences and trees. In *IJCAI*.

Knight, K., and Graehl, J. 2004. Training tree transducers. In *HLT-NAACL*. Association for Computational Linguistics.

Melamed, I. D.; Satta, G.; and Wellington, B. 2004. Generalized multitext grammars. In *ACL*.

Och, F. J., and Ney, H. 2000. Improved statistical alignment models. In *ACL*.

Wu, D. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist.* 23(3).

Zhang, H., and Gildea, D. 2006. Efficient search for inversion transduction grammar. In *EMNLP*.

[3] All inside and outside costs are Viterbi, not summed.

[4] The stated objective is merely one reasonable choice among many possibilities which requires pointwise admissibility and encourages tight estimates.