Search-based Learning

Michael Collins and Brian Roark. *Incremental Parsing with the Perceptron Algorithm*.

Hal Daume III, Daniel Marcu. *Learning as search optimization: Approximate Large Margin Methods for Structured Prediction*

Hal Daume III, John Langford, Daniel Marcu. *Search-Based Structured Prediction.*

Presented by Veselin Stoyanov

Structured Learning

- In the heart of all algorithms so far:

$$y^* = \arg\max_{y \in Y} f(x, y, w)$$

- I.e., an exhaustive search over all y
- This computation can be very expensive or computationally impossible

When the *argmax* is intractable…

- E.g. joint label+sequence inference, parsing, multi-document compression
- An idea:
  - Take advantage of the sequential nature of the decision
  - Decompose the problem and limit the search space

Approach 1: Collins and Roark

Incremental Parsing with the Perceptron Algorithm

- Reminiscent of the Structured Perceptron algorithm

Given (x_i, y_i)
For t = 1,..T,i = 1,..n
  $z_i = \text{argmax}_{z \in Gen(x_i)} f(x_i,z,w)$
  if$(z_i != y_i)$ then $w = w+\Phi(x_i,y_i)-\Phi(x_i,z_i)$

Approach 1: Collins and Roark

Incremental Parsing with the Perceptron Algorithm

Noun Verb Adj. Noun

Yejin likes merino sheep.

Approach 1: Collins and Roark

Incremental Parsing with the Perceptron Algorithm

Yejin likes merino sheep.

Approach 1: Collins and Roark
Incremental Parsing with the Perceptron Algorithm

Noun	Noun	Noun	Noun
Verb	Verb	Verb	Verb
Adj.	Adj.	Adj.	Adj

Yejin likes merino sheep.

Approach 1: Collins and Roark
Incremental Parsing with the Perceptron Algorithm

Noun	Noun	Noun	Noun
Verb	Verb	Verb	Verb
Adj.	Adj.	Adj.	Adj

Yejin likes merino sheep.

Approach 1: Collins and Roark
Incremental Parsing with the Perceptron Algorithm

Noun
Verb
Adj.

Yejin likes merino sheep.

Approach 1: Collins and Roark
Incremental Parsing with the Perceptron Algorithm

Noun Noun
Noun Verb
Noun Adj.
Verb Noun
Verb Verb
Verb Adj.
Adj. Noun
Adj. Verb
Adj. Adj.

Yejin likes merino sheep.

Approach 1: Collins and Roark
Incremental Parsing with the Perceptron Algorithm

Noun
Verb
~~Adj.~~

Yejin likes merino sheep.

Approach 1: Collins and Roark
Incremental Parsing with the Perceptron Algorithm

~~Noun Noun~~
Noun Verb
~~Noun Adj.~~
Verb Noun
~~Verb Verb~~
Verb Adj.

Yejin likes merino sheep.

Approach 1: Collins and Roark
Incremental Parsing with the Perceptron Algorithm

Noun Verb
Verb Noun
Verb Adj.

Yejin likes merino sheep.

Approach 1: Collins and Roark
Incremental Parsing with the Perceptron Algorithm

- New incremental Perceptron algorithm

Given (x_i, y_i)
For t = 1,..T,i = 1,..n
$z_i = \text{argmax}_{z \in F(x_i)} f(x_i,z,w)$
if$(z_i != y_i)$ then $w = w+\Phi(x_i,y_i)-\Phi(z_i,y_i)$

Approach 1: Collins and Roark
Incremental Parsing with the Perceptron Algorithm

- Refinements to the learning algorithm
  - Repeated use of hypothesis
    - Construct the F-sets and use them for a few iterations of the perceptron update
  - Early update
    - If the correct labeling falls out of the beam, stop decoding and use the partially decoded sequence

Approach 1: Collins and Roark
Incremental Parsing with the Perceptron Algorithm

- Experimental evaluation: Parsing (Pen Treebank)
- Results:
  - With standard features identical to generative model
  - Using the output of the generative model as a feature shows improvement (f1 of 88.8 vs. 86.7)

Approach 2: Daume and Marcu:
Learning as Search Optimization

- Similar to Collins and Roark
- Theoretical contributions:
  - Generalizes learning as search
  - Gives theoretical justifications of the algorithm
- Differences in the parameter update:
  - When gold standard falls out of the beam:
    - Update parameters and continue from correct solutions
  - Approximate maximum margin update

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- Advances the idea one step further
- Keep the beam size 1
  - i.e. at every step, predict the best output given the partial output so far
- Thus, at every step you have a multiclass classification problem

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- Removes the search from the process
- Can incorporate any multiclass classifier
- Can handle more general features and loss function
- Theoretically sound
  - Proofs that good performance on the multiclass classification leads to good performance on the structured prediction

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

Yejin likes merino sheep.

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

Noun

Yejin likes merino sheep.

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

Noun Verb

Yejin likes merino sheep.

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

Noun Verb Adj.

Yejin likes merino sheep.

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

Noun Verb Adj. Noun

Yejin likes merino sheep.

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- Problem: Once the classifier makes an error (exits the Garden Path) performance deteriorates
- Solution: Start with the optimal policy and slowly move to a learned policy

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- A few definitions:
  - – Policy
    - A policy *h* is a distribution over actions conditioned on an input *x* and state *s.*
    - (The multiclass prediction)
  - – Optimal policy
    - A policy that, for a given state, input and output always predicts the best action to take.
    - π*

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- A few definitions:
  - – Cost-sensitive examples

$$l_a^{\pi} = \mathrm{E}_{y \sim search(x_n, \pi, a)} c_y - \min_{a'} l_a^{\pi}$$

  - – The loss for every state (partially decoded sequence) and every possible action
  - – The loss for the action = regret for taking this action vs. the optimal action
  - – Depends on the policy π

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- Cost-sensitive examples:
  - – How are they computed
    - Given a state s
    - For every possible action a
      - – Compute the expected loss of the action by decoding the full sequence using the current policy π
    - Create a cost vector for state s

    (l_{a1}^{π}- min l_a^{π}; l_{a2}^{π}- min l_a^{π} ;…; l_{am}^{π}- min l_a^{π})
    - These costs will be used to train the multiclass classifier

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- The algorithm

$$
\begin{array}{l}
h^{(0)} \leftarrow \pi^* \\
\mathit{for}\ I = 1 \ldots I_{\max}\ \mathit{do} \\
\quad S_I \leftarrow 0 \\
\quad \mathit{for}\ n = 1 \ldots N\ \mathit{do} \\
\qquad \left\langle s_1, \ldots, s_{T_n} \right\rangle \leftarrow pth\left(x_n, h^{(I-1)}, \mathbf{0}\right) \\
\qquad Add\left(\Phi(x_n, s_t), l_{s_t \oplus 1}^{h^{(I-1)}}, \ldots, l_{s_t \oplus |A|}^{h^{(I-1)}}\right) to\ S_I \\
\quad \mathit{end\ for} \\
\quad h' \leftarrow Learn(S_I) \\
\quad h^{(I)} \leftarrow \beta h' + (1-\beta) h^{(I-1)} \\
\mathit{end\ for} \\
\mathit{return}\ h^{(I_{\max})} - \pi^*
\end{array}
$$

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- Theorem:

$$L(D,h) \le L(D,h_0) + CTl_{avg} + c_{\max}\left(\frac{1}{2}CT^2\beta + T\exp[-C]\right)$$

$$C = 2\ln T; \quad \beta = 1/T^3$$

$$L(D,h) \le L(D,h_0) + 2T\ln Tl_{avg} + \left(1+\ln T\right)c_{\max}/T$$

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- Computational requirements
  - Optimal policy assumption

$$\arg\min_{y \in Y_x} w^T \Phi(x, y^*) + l(y, y^*)$$

  - Additionally, optimal policy is not required

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- Beyond greedy search
  - Could add the beam search back in the algorithm
- Similar to reinforcement learning

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- Beyond greedy search
  - Could add the beam search back in the algorithm
- Similar to reinforcement learning

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- Empirical evaluation
  - Spanish named entity recognition
  - Handwriting recognition
  - Joint sequence labeling
  - Multi-document compression

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- Spanish named entity recognition

	CRF	SVM	SVM ISO	SEARN	
				300	All
Acc	94.83	94.94	94.90	**95.01**	**97.67**

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- Handwriting recognition

	M^3N			SEARN				
	d1	d2	d3	Perc	LR	SVM-d1	SVM-d2	SVM-d3
Acc-sm	81.00	87.00	87.50	70.17	73.81	82.12	87.55	88.20
Acc-lg	-	-	-	76.88	79.28	90.58	**90.91**	90.22

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- Joint sequence labeling

	Joint Acc	Chunk F-score	POS Acc
SEARN			
Joint Acc (G)	95.09	92.99	98.90
Chunk F (G)	-	93.60	-
Joint Acc (B)	**96.81**	93.62	**99.07**
Chunk F (B)	-	**94.63**	-
Incr. Perceptron (G)	91.40	90.82	96.94
Incr. Perceptron (B)	93.12	92.44	98.05
LaSO (G)	94.74	93.11	98.89
LaSO (B)	95.12	93.49	98.90
Factored CRF	**96.48**	93.87	**98.92**
CRF	-	**94.77**	-

Approach 3: Daume, Langford and Marcu:
Search-based Structured Prediction

- Multi-document compression

	100 words	**200 words**
Oracle		
Vine-Growth	7.32	22.72
Extraction	4.41	13.61
SEARN		
Vine-Growth	4.98	**15.80**
Extraction	4.28	12.33
BQFS+DUC05	3.97	**10.89**
BQFS+DUC03	3.54	9.47
Baseline	2.41	5.82
Best DUC05	-	**10.19**

Summary

- Collins and Roark's *Incremental Perceptron*
  - Incremental decoding and perceptron-style learning
- Daume and Marcu's *LaSO*
  - Generalize the search procedure
  - Theoretical justification
  - Approximate max-margin update
- Daume, Langford and Marcu's *SEARN*
  - Separate incremental learning and the perceptron
  - Can incorporate any multiclass classifier
  - Can handle more general features and loss function
  - Theoretically sound