Lecture 23 - L2S; SeaRNN; SPENs

Tuesday, April 11, 2023 2:29 PM

today: • learning to search
• SEARNN
• SPENs

learning to search (L2S)

SEARN Hal Daumés phd thesis

$h_w : \mathcal{X} \to \mathcal{Y}$

$h_w(x) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \; s(x, y; w)$] parameterized search procedure

today: special case of SEARN: learning to do greedy search

split y in ordered list of decisions

$(y_1, y_2, y_3, \ldots, y_T)$

learn a classifier $\pi_w(\text{feature}(\hat{y}_1, \ldots, \hat{y}_{t-1}; x)) = \hat{y}_t$

↑ classifier "decoding policy"

L2S framework

from $(x^{(i)}, y^{(i)})_{i=1}^n$ and $\ell(\cdot, \cdot)$

learn a good classifier/policy $\pi_{\hat{w}}$ s.t. $\hat{y}_t = \pi_{\hat{w}}(\varphi(\hat{y}_1, \ldots, \hat{y}_{t-1}; x))$

$h_{\hat{w}}(x) \triangleq (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T)$ "greedy decoder"

s.t. $\ell(y^{(i)}, h_{\hat{w}}(x^{(i)}))$ is small

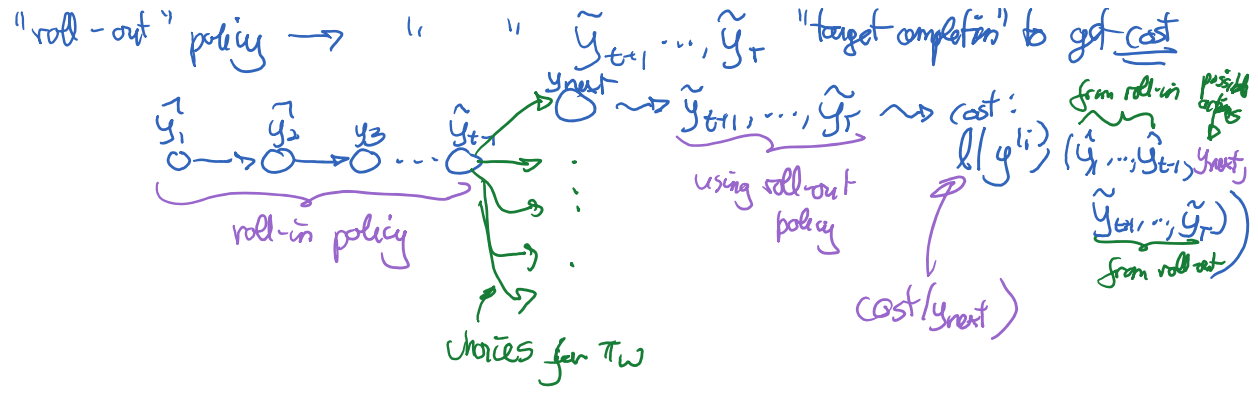
⭐ central idea: "reduction" that reduces the structured prediction problem to sequence of cost sensitive classification learning of π_w

method: generate training data for classifier π_w

$(\hat{y}^{(i)}_{\text{context}}, x^{(i)}, \text{cost}(y_{\text{next}}))$

prefix sequence to make next prediction

"roll in" policy → determines how $\hat{y}_{1:t-1}$ context



"reference policy", ideally, $\pi_{ref}((\hat{y}_1, \dots, \hat{y}_{t-1}, y_{next}))$

$= \underset{y_{completion}}{\operatorname{argmin}} \ell(y^{(i)}, (\hat{y}_{1:t-1}, y_{next}, y_{completion}))$

intractable (NP hard) in general

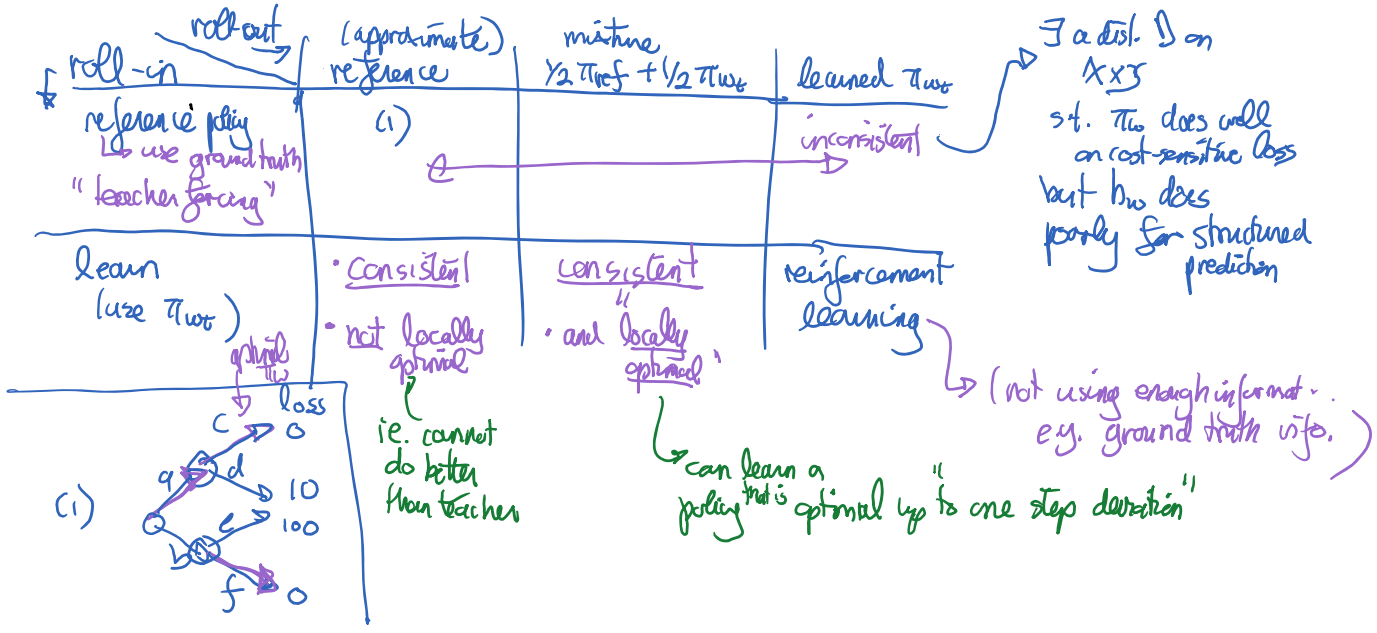
in practice: use heuristics to approximate π

but sometimes, can compute exactly

e.g. π_{ref} for Hamming loss: is just copy ground truth

ICML 2015 "L2S better than your teacher"

LOLS → "locally optimal learning to search"



example of approximate π_{ref} (in machine translation): ℓ is bleu score

consider all possible suffixes from ground truth and pick the best bleu score one

15h30

SEARNN : apply L2S to RNN training

[Leblond & al. ICLR 2018] ie $\pi_w(y_{1:t-1}, x) \rightsquigarrow$ RNN cell

$p(y_{next} \mid y_{1:t-1}, x)$ of a (decoder) RNN

if you use $-\log p_w(\hat{y}_{target} \mid y_{1:t-1}, x)$ as a cost surrogate

and $\hat{y}_{target}$ = ground truth

then L2S with π_{ref} roll-in is standard MLE

cost-sensitive surrogate losses choices

a) structured SVM : $\max_{\tilde{y}} [\underbrace{\Delta(\tilde{y})}_{\triangleq c(\tilde{y}) - c(y_{target})} + s(\tilde{y})] - s(y_{target})$

$y_{target} \triangleq \arg\min_{y'} c(y')$

b) "target log-loss" : $-\log p_w(\hat{y}_{target} \mid y_{1:t-1}, x)$

↳ differences with MLE :
- address exposure bias using learned roll-in
- make use of structured loss $\ell(\cdot,\cdot)$ to predict $\hat{y}_{target}$ vs. ground truth for MLE

structured prediction energy networks (SPENs)

ICML 2016

$E(x, y; w)$ $\quad (-s(x, y; w))$

- relax $y \in \{0,1\}^T \rightsquigarrow [0,1]^T$

$h_w(x)$ = a few steps of projected G.D. on $E(x, y; w)$ w.r. to y + rounding (approximate prediction procedure)

in 2016 paper : SSVM loss $\rightsquigarrow$ "subgradient" method on w

hinge loss : $\max_{\bar{y} \in [0,1]^T} \left(\ell(y^{(i)}, \bar{y}) - E(x^{(i)}, \bar{y}; w) \right) + E(x^{(i)}, y^{(i)}; w)$

requires extending ℓ to fractional y's

augmax $\triangleq \bar{y}^*$

approximate "subgradient" is $-\nabla_w E(x^{(i)}, \bar{y}^*(w_t); w_t) + \nabla_w E(x^{(i)}, y^{(i)}; w)$

because use $\bar{y}^*$ as approximate max

e.g. see Clarke subdifferential for generalization on non-convex fct.

$y_1 = y_0 - \eta_1 \frac{d}{dy} E(x, y_0; w)$

$y_2 = y_1 - \eta_2 \frac{d}{dy} E(x, y_1; w)$

etc...

recursively defined

2017 paper : end-to-end learning

$$h_w(x) = y_0 - \sum_{t=1}^{T} \eta_t \frac{d}{dy} E(x, y_t; w)$$

"unrolled optimization"