Lecture 23 - L2S, SEARNN

Tuesday, April 7, 2020 14:21

today: • learning to search
• SEARNN
• SPEN

Learning to search (L2S)

SEARN Hal Daume's 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

special case of SEARN: learning to do greedy search

split y in ordered list of decisions

$(y_1, \dots, y_T)$

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

classifier "decoding policy"

L2S framework

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

learn a good classifier/policy π_w s.t. $\hat{y}_t = \pi_w(\phi(\hat{y}_1, \dots, \hat{y}_{t-1}, x))$

$h_w(x) \triangleq (\hat{y}_1, \dots, \hat{y}_T)$ "greedy decoder"

s.t. $\ell(y^{(i)}, h_w(x^{(i)}))$ is good

central idea: "reduction" where reduces structured prediction learning to problem of cost sensitive classification learning for π_w

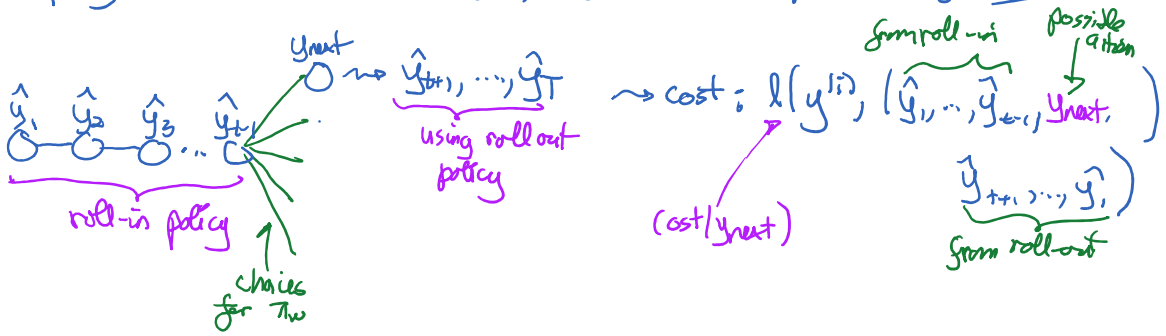
method: generate training data for classifier π_w

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

prefix sequence

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

"roll-out" policy → " " " $y_{t+1}, \dots, y_T$ "target completion" to get cost



"reference policy" ideally, $\pi_{ref}(\hat{y}_1, \dots, \hat{y}_{t-1}, y_{next}) = \text{argmin}_{y_{completion}} l(y^{[i]}, (\hat{y}_{1:t-1}, y_{next}, y_{completion}))$

intractable (NP hard) in general

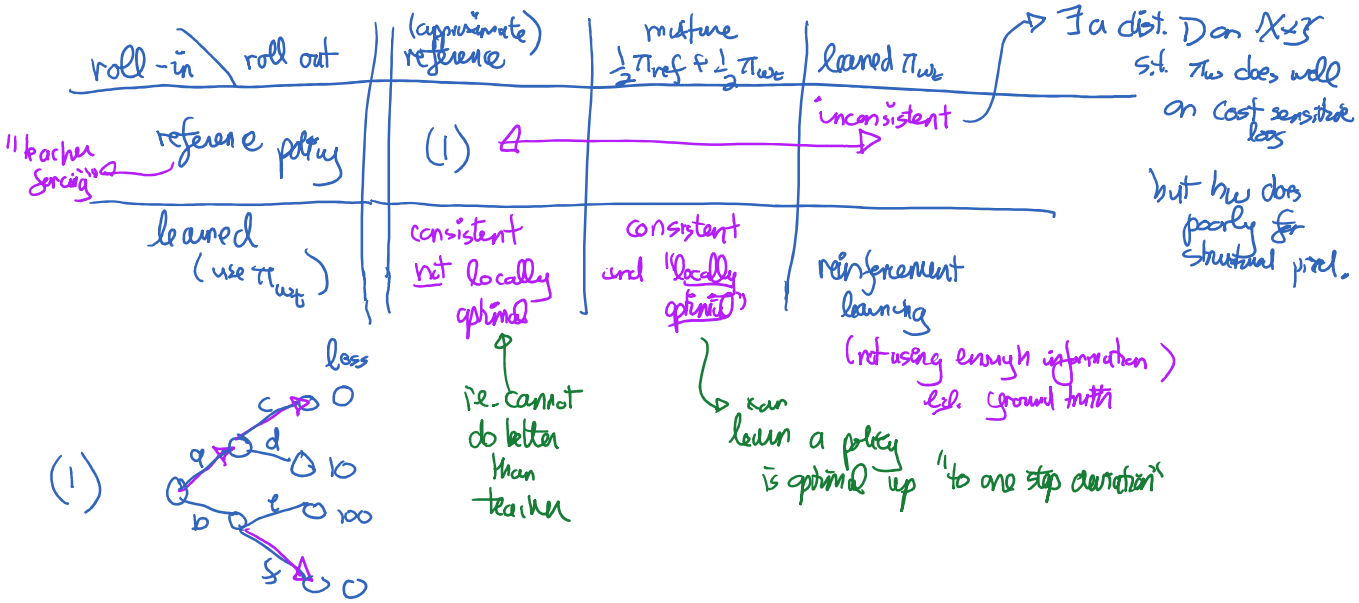
in practice: use heuristic to approximate π

but sometimes can compute exactly

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

LOLS → "locally optimal learning to search"

ICML 2015 "Learning to search better than your teacher"



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

consider all possible suffixes and pick

the best BLEU-1 score

15h35

SEARNN : apply L2S to RNN training

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

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

If use $-\log p(\hat{y}_{target} | y_{1:t-1}, x)$ as cost surrogate

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

then L2S with ref roll-in is standard MLE

③ Cost-sensitive surrogate loss choices:

a) structured SVM: $\max_{y'} [\Delta(y') + s(y')] - s(y_{target})$

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

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

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

↳ differences with MLE:

- address exposure bias using learned roll-in
- make use of structured loss $\ell(\ldots)$ to predict $\hat{y}_{target}$ vs. MLE

Kushal Arora: How (not) to Train your Generative Model: Scheduled Sampling, Likelihood, Adversary? by Huszar answers this question why SS might not work well. This is what I observed as well.

Structured prediction energy networks (SPENs)

ICML 2016

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

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

$h_w(x)$ = a few steps of projected G.D. on $E(x, y; w)$ w.r.t. y / approximate prediction

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

2016 paper:

SSVM loss → "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 an extension

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

$\bar{y}^*$ is approx max.

e.g. see Clarke subdifferential

2017 paper: end-to-end learning:

recursively define

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

"unrolled optimization"