Lecture 23 - L2S, SEARNN

Tuesday, April 9, 2019 14:37

today: • learning to search
• SEARNN
• SPEN

Learning to search (L2S)

SEARN Hal Daume's phd thesis

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

$h_w(x) = \arg\max_{y \in \mathcal{Y}} s(x, y; w)$

special case: learning to do greedy search

split y in ordered list of decisions

$(y_1, \ldots, y_T)$

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

π: classifier deciding "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

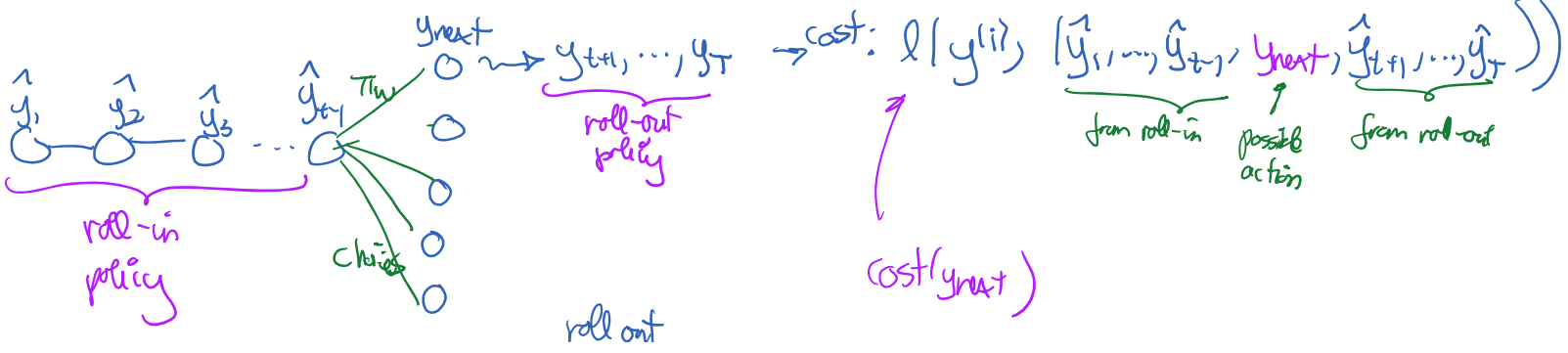
method: generate training data for classifier π_w

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

$\hat{y}^{(i)}_{\text{context}}$: prefix sequence

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

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



"reference policy"

ideally: $\pi_{ref}(\hat{y}_1, \ldots, \hat{y}_{t-1}, y_{next})$

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

intractable (NP-hard) in general

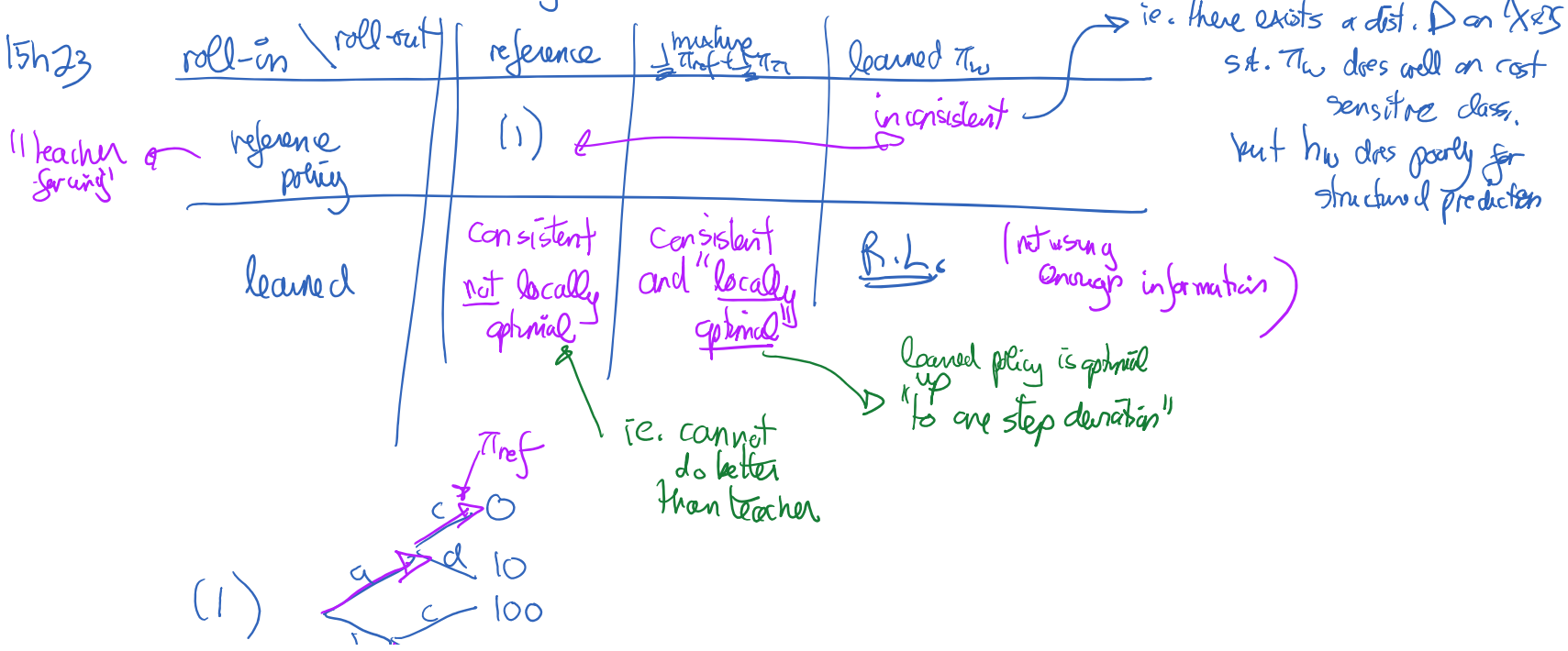
in practice: heuristic & approximate it

but sometimes can compute exactly

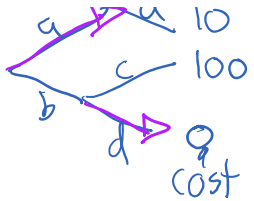
e.g. π_{ref} for Hamming loss is just copy ground truth

LOLS → "locally optimal learning to search"

ICML 2015 "LOLS better than you teacher"



(1)



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

consider all possible g.t. suffices and pick the best BLEU-1 scoring

SEARNN: apply L2S to RNN training

ie. $\pi_w(y_{1:t-1}, x) \rightsquigarrow$ RNN cell

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

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

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

then L2S with ref rollin is standard MLE

(2) cost-sensitive surrogate choices:

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

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

$y_{target} = \text{argmin}_{y'} c(y')$

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

↳ differences with MLE:

- address the exposure bias using learned roll-in
- make use of structured loss $\ell(\dots)$ to pick $\hat{y}_{target}$ vs. 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 gradient descent on $E(x, y; w)$ with respect to 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 convexity

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

e.g. see Clarke subdifferential

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"