Lecture 22 - latent SVMstruct; RNN

Tuesday, April 6, 2021 14:27

today : · latent variable SVMstruct — CCCP
· deep learning — RNN

latent variables

motivation : semantic segmentation

segmentation is expensive → z "latent variable"

perhaps only have class labels → y

also : [Felzenszwalb & al TPAMI 2010]
"deformable part models" for object recognition
↳ z there was an object part configuration

before, we had $s(x, y; w) = \langle w, \varphi(x, y) \rangle$

now, consider $s(x, y, z; w) = \langle w, \varphi(x, y, z) \rangle$

as before, could predict with $\underset{y \in \mathcal{Y}, z \in \mathcal{Z}}{\operatorname{argmax}} \; s(x, y, z; w)$

learning — CAF approach $p(y|x)$ → generalize → hidden CAF $p(y, z|x)$

similarly to latent variable modeling with graph model
→ marginalize z out

ML → EM (expectation-maximization)

analog for latent SVMstruct is CCCP (majorization minimization procedure)

SVMstruct (surrogate loss minimization)

latent SVMstruct

$\ell(y, (\tilde{y}, \tilde{z}))$

generalize structured hinge loss

best score for ground truth

$$g(x, y, w) \triangleq \underbrace{\max_{\tilde{y}, \tilde{z}} \langle w, \varphi(x, \tilde{y}, \tilde{z}) \rangle + \ell(y, (\tilde{y}, \tilde{z}))}_{\triangleq u(w)} - \underbrace{\max_{z' \in \mathcal{Z}} \langle w, \varphi(x, y, z') \rangle}_{\triangleq v(w)} \geq \ell(y, h_w(x))$$

$(\tilde{y}, \tilde{z})$

$\max_{\tilde{y},\tilde{z}}$ ($\triangleq u(w)$) $\qquad$ $\max_{z' \in Z}$ ($\triangleq v(w)$) $\qquad$ $(\hat{y}, \hat{z})$

here $f(x,y;w) = u(w) - v(w)$ where u & v are convex fct. of w

"difference of convex functions"

↳ CCCP procedure is to approx. minimize this

CCCP procedure:

- linearize $v(w)$ at w_t to get an upper bound
- w_{t+1} is obtained by minimizing this upper bound
- repeat ↳ a majorization-minimization procedure (EM is another example)

$$\begin{cases} f_t(w) = u(w) - [v(w_t) + \langle \nabla v(w_t), w - w_t \rangle] \geq f(w) \ \forall w \\ \qquad \text{and } f_t(w_t) = f(w_t) \\ w_{t+1} = \operatorname{argmin}_w f_t(w) \end{cases}$$

($\nabla v(w_t)$ (or subgradient))

properties of procedure:

- like EM, descent procedure ie. $f(w_{t+1}) \leq f(w_t)$

$$f(w_t) = f_t(w_t) \geq f_t(w_{t+1}) \overset{\text{upper bound}}{\geq} f(w_{t+1}) \quad //$$

- local linear convergence to a stationary pt. for latent SVMstruct [see NeurIPS OPT 2012 paper]

---

* **CCCP for SVMstruct:** $v(w) = \max_{z'} \langle w, \varphi(x,y,z') \rangle$

$\triangleq \operatorname{argmax}_{z'} \langle w_t, \varphi(x,y,z') \rangle$

$\partial v(w_t) = \varphi(x, y, \hat{z}(x,y,w_t))$

$$\Rightarrow f_t(w) = \max_{(\tilde{y},\tilde{z})} \langle w, \varphi(x,\tilde{y},\tilde{z}) \rangle + \ell(y,(\tilde{y},\tilde{z})) - \langle w, \varphi(x,y,\hat{z}_t) \rangle + \text{cst.}$$

$\rightsquigarrow$ like SVMstruct objective

CCCP algorithm for latent SVMstruct:

repeat:
- fill in $\hat{z}_t^{(i)}$ for all ground truth $y^{(i)}$ using w_t
- solve a standard SVMstruct to get w_{t+1}

repeat:
- fill in z_t for all ground truth $y^{(i)}$ using w_t
- solve a standard SVM struct to get w_{t+1}
- repeat

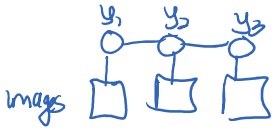
Deep learning

can learn

go from $\langle w, \varphi(x,y) \rangle$ to $\langle w, \varphi(x,y;\theta) \rangle$

I) plug in "deep learning" features in a structured prediction model

example: OCR



so far $\varphi_t(x_t, y_t) = \begin{pmatrix} 0 \\ x_t \\ 0 \end{pmatrix} \leftarrow y_t^{th}$

instead $\varphi_t(x_t, y_t) = \begin{pmatrix} 0 \\ NN_\theta(x_t) \\ 0 \end{pmatrix} \leftarrow y_t^{th}$ → pre-trained on imagenet e.g.

example: [Vu et al. ICCV 2015] "context-aware CNNs for person head detection"

II) "end-to-end" training

structured prediction energy networks (SPENs)

III) recurrent neural networks (RNN)

motivation: $p(y|x) = \prod_{t=1}^{T} p(y_t | y_{1:t-1}, x)$ ← chain rule

graphical modeling approach → $\prod_t p(y_t | y_{\pi_t}, x)$ ($y_t \perp y_{1:t-1} | y_{\pi_t}$ directed)

or $\frac{1}{Z(x)} \prod_{c \in C} \psi_c(y_c; x)$ [undirected]

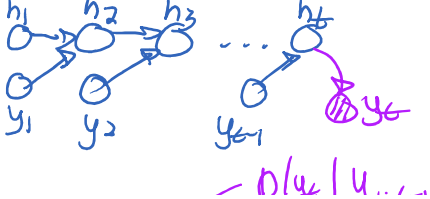
RNN → "structured parameterization" of $p(y_t | y_{1:t-1}, x)$ using NN

with no cond. indep assumptions → usually loose → exact decoding

$$h_{t+1} \triangleq f(h_t, x, y_t, W)$$

$$h_t = f(f(\dots (h_1, x, y_1, W), x, y_2) \cdots), x, y_{t-1}, W)$$

define $p(y_t | y_{1:t-1}, x) \propto \exp(c(y_t)^T \tilde{W} h_t)$ ("code" e.g.)



$= p(y_t | y_1, \dots, x)$

$p(y_t \mid y_{1:t-1}, x) \propto \exp(\ldots)$ $\bar{y}_1$ $\bar{y}_2$ $\bar{y}_{t-1}$ → $p(y_t \mid y_{1:t-1}, x)$

e.g. "code" for y_t

given by a deep NN architecture

e.g. word embedding in NMT and can be fine tuned

Standard learning: using ML

ie. $\min_{W, \tilde{W}} -\frac{1}{n} \sum_{i=1}^{n} \log p(y^{(i)} \mid x^{(i)})$

$\sum_t \log p(y_t^{(i)} \mid y_{1:t-1}^{(i)}, x^{(i)})$

output of a deep NN

"teacher forcing" → "exposure problem" ie. don't know $p(\tilde{y}_t \mid \text{unseen}, x, y_{1:t-1})$

14h43

for ML, do SGD objective

gradient of $\log p(y_t^{(i)} \mid y_{1:t-1}^{(i)}, x^{(i)}; W, \tilde{W})$ → use back propagation

decoding: $\operatorname{argmax}_{y \in \mathcal{Y}} \sum_t \log p(y_t \mid y_{1:t-1}, x)$ → NP hard!

→ need approximation — greedy decoding $\hat{y}_t = \operatorname{argmax}_{y_t \in \mathcal{Y}_t} p(y_t \mid \hat{y}_{1:t-1}, x)$

beam search "greedy decoding with memory of size k" → size of beam

beam search: construct $\hat{y}_1, \ldots, \hat{y}_T$

beam of size L (memory)

- at step t, you have L candidate solution prefixes $y_{1:t}^{(1)}, \ldots, y_{1:t}^{(L)}$
- expand possible next choice $|\mathcal{Y}_{t+1}| \cdot L$

score them (e.g. $\log p(y_{t+1} \mid \hat{y}_{1:t}^{(l)}, x) + \log p(\hat{y}_{1:t}^{(l)}, x)$)

then keep top L candidates as $\{y_{1:t+1}^{(l)}\}_{l=1}^{L}$

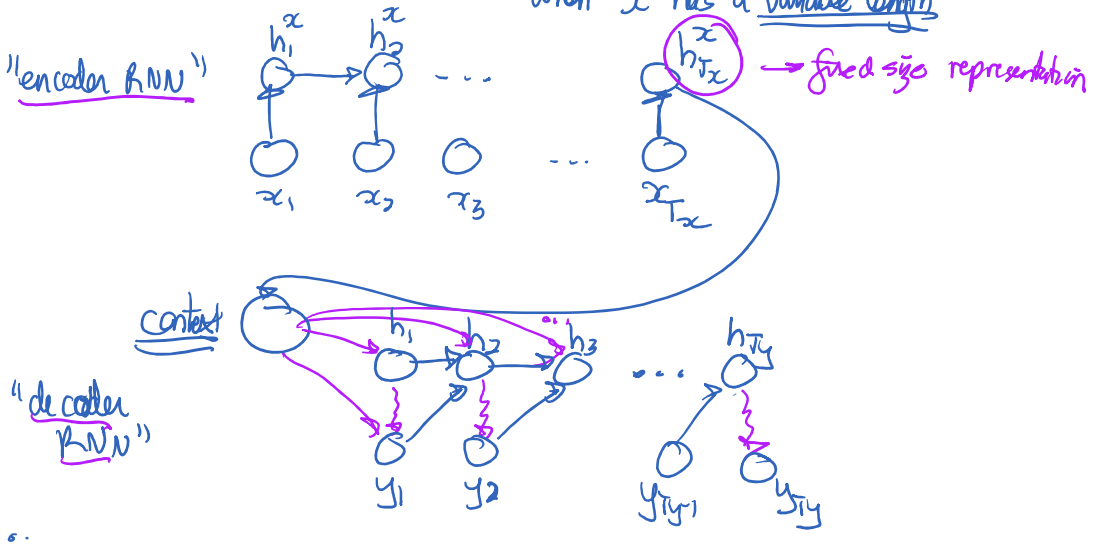
Vs.

Viterbi alg. which does "backtracking" to correct past mistakes

Seq2seq a.k.a. encoder/decoder architecture

↳ useful way to get $p(y_t \mid y_{1:t-1}, x)$ for a RNN

when x has a variable length



Issues:

a) variable length output? → end-of-sequence character

b) long input sequence x? (how to handle)

problem: need to summarize input sentence in one context vector of fixed length

Solution: "attention mechanism"

c) vanishing gradient?

- LSTM
- gated recurrent unit (GRU)
- etc..