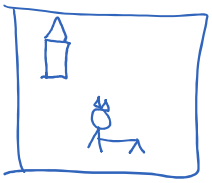
lecture 22 - latent SVMstruct; RNN

Thursday, April 2, 2020 13:25

today: • latent variable SVMstruct – CCCP
• deep learning

latent variables

motivation: Semantic segmentation → find boundary of different objects



segmentation is expensive → z "latent variable"

perhaps only have class labels → y

also: [Felzenszwalb & al TPAMI 2010]
"deformable part models" for object recognition
↳ z these were an object part configurations

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

(*) CRF $(p(y|x))$ ~~generalize~~→ hidden CRF $p(y, z|x)$

similar to latent variable modeling with graph. model

ML ⇝ EM (expectation-maximization)

↳ analog for latent SVMstruct is CCCP

latent SVMstruct

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

$w(w)$: concatenation of w

generalize structured hinge loss:

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

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

$$g(x,y,w) = \max_{\tilde{y},\tilde{z}} \langle w, \varphi(x,y,z)\rangle + \ell(y,(\tilde{y},\tilde{z})) - \max_{z' \in Z} \langle w, \varphi(x,y,z')\rangle \geq \ell(y,h_w(x))$$

$\triangleq v(w)$; $(\tilde{y},\tilde{z})$

here $g(x,y,w) = u(w) - v(w)$ 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)

$$g_t(w) = u(w) - [v(w_t) + \langle \nabla v(w_t), w - w_t\rangle] \geq g(w) \ \forall w$$

(∇ or subgradient) and $g_t(w_t) = g(w_t)$

$$w_{t+1} = \operatorname*{argmin}_w g_t(w)$$

properties of procedure:

- like EM, descent procedure i.e. $g(w_{t+1}) \leq g(w_t)$

$$g(w_t) = g_t(w_t) \geq g_t(w_{t+1}) \geq g(w_{t+1})$$

(upper bound)

- local linear convergence to a stationary point [see NIPS OPT 2012 paper] for latent SVMstruct

---

* for SVMstruct: $v(w) = \max_z \langle w, \varphi(x,y,z)\rangle \triangleq \operatorname{argmax}_z \langle w_t, \varphi(x,y,z)\rangle$

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

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

↝ like SVMstruct objective

CCCP algorithm for latent SVMstruct:

repeat: • fill in $\hat{z}_t^{(i)}$ for all ground truth $y^{(i)}$ using w_t

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

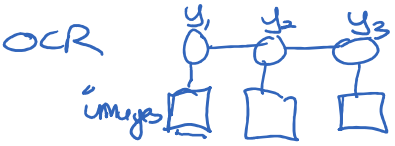
Deep learning

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 $\varphi_t(x_t, y_t) = \begin{pmatrix} 0 \\ NN_\theta(x_t) \\ 0 \end{pmatrix} \leftarrow y_t$

learned on imagenet e.g.

example: [Vu & 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)

$y_t \perp\!\!\!\perp y_{1:t-1} \mid y_{\pi_t}$

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

graphical model approach

[directed] $\prod_t p(y_t | y_{\pi_t}, x)$

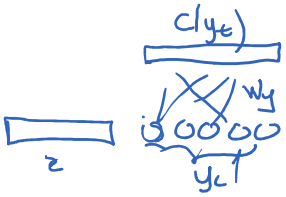
[undirected] $\frac{1}{Z(x)} \prod_{c \in \mathcal{C}} \psi_c(y_c, x)$

RNN → "structured parameterization" of $p(y_t | y_{1:t-1}, x)$ with no cond. indep. assumptions

using NN

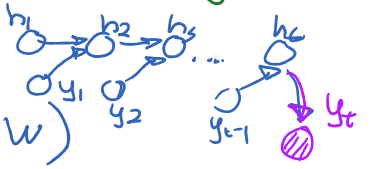
↳ usually loose

exact decoding



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

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



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

$p(y_t | y_{1:t-1}, x)$ ↓ given by (deep) NN architecture

standard learning: use M.L.

$\min \frac{1}{n} \sum_{i=1}^{n} \ell(\dots | \dots)$

Standard learning: use M.L.

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

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

output of a deep NN

"teacher forcing"

exposure problem

ie; do not know $p(\tilde{y}_t | \text{unseen } x)$ prefix $\tilde{y}_{1:t-1}$

14h41

for ML, do SGD objective

gradient of $\log p(y_t^{(i)} | y_{1:t-1}^{(i)}, x^{(i)}; W, \tilde{W})$

↝ use backpropagation

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

↝ need approximation ← greedy decoding $\hat{y}_t = \underset{y_t \in \mathcal{Y}_t}{\text{argmax}}\, p(y_t | \hat{y}_{1:t-1}, x)$

beam search "greedy decoding with memory of size K"

"beam"

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

beam of size L (memory)

- at step t, you have L candidate solution prefixes $\hat{y}_{1:t}^{(1)}, \dots, \hat{y}_{1:t}^{(L)}$
- expand possible next choice: $|\mathcal{Y}_{t+1}| \cdot L$
- score them $(\text{e.g } \log p(y_{t+1} | \hat{y}_{1:t}^{(l)}) + \log p(\hat{y}_{1:t}^{(l)}))$

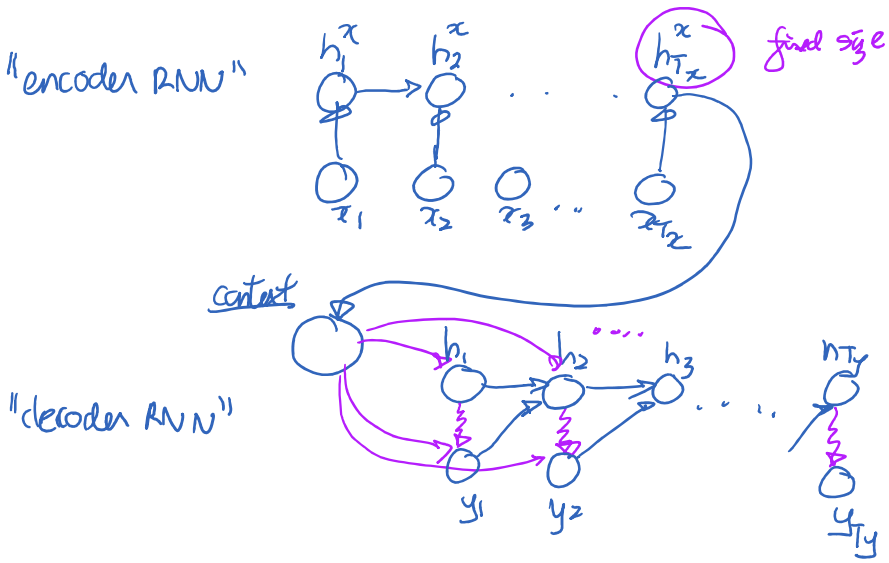
  then keep top L candidates as $\{\hat{y}_{1:t+1}^{(l)}\}_{l=1}^{L}$

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

seq2seq aka encoder/decoder architecture

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

When x has variable length



Issues:

a) variable length output?

→ end-of-sequence sequence character

b) long input sequence x?

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

solution: "attention mechanism"

c) vanishing gradient?

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