Lecture 22 - latent SVMstruct

Thursday, April 4, 2019 13:33

today : • latent variable SVMstruct ~ CCCP
• deep learning

prox SAGA

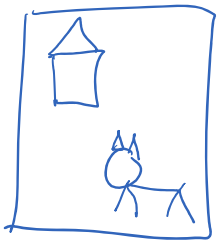
$$\min_w \frac{1}{n}\sum_i f_i(w) + \Omega(w)$$

$$w_{t+1} = \text{Prox}_{\gamma}^{\Omega}\left(w_t - \gamma\left[\nabla f_{i_t}(w_t) - g_{i_t}^t + \frac{1}{n}\sum_j g_j^t\right]\right)$$

→ what is used by default in Scikit-learn for lasso

latent variables

motivation : semantic segmentation → find boundary of different objects



segmentation is expensive → z "latent variable"

perhaps only have class labels → y

also : [Felzenszwalb et al. TPAMI 2010

"deformable part models" for object recognition

↳ z : where was an object

conjugation

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

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

similar to latent variable modeling with graph. model

ML ⇝ expectation-maximization (EM)

↳ analog for latent SVMstruct is CCCP

Latent SVMstruct :

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

generalization of structured hinge loss:

concave fct. of w

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

$(\tilde{y},\tilde{z})$ ↑

here $\delta(x,y,w) = u(w) - v(w)$ where u & v are convex fcts of w

"difference of convex functions"

↳ CCCP procedure is used to approx minimize this

CCCP procedure

- linearize $v(w)$ at w_t to get an upper bound
- minimize the upper bound
- repeat

↳ a majorization-minimization procedure (EM is another example)

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

($\nabla v(w_t)$: a subgradient)

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

properties of this procedure:

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

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

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

* for SVMstruct

$$v(w) = \max \langle w, \varphi(x, y, z) \rangle$$

$$V(w) = \max_z \langle w, \varphi(x,y,z)\rangle$$

$$\partial V(w_t) = \varphi(x, y, \hat{z}(x, w_t)) \quad \text{where } \hat{z}(x,w_t) = \operatorname{argmax}_z \langle w_t, \varphi(x,y,z)\rangle$$

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

$\rightsquigarrow$ like SVMstruct objective

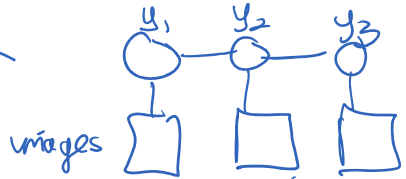
CCCP for latent SVMstruct: repeat:
- fill in $\hat{z}_t^{(i)}$ for all ground truth $y^{(i)}$ using w_t
- solve standard SVMstruct to get w_{t+1}
- repeat

Deep learning

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

I) plug in "deep learning" features in a structured predict. model

example: OCR



this class: $\varphi_t(x_t, y_t) = \begin{pmatrix} 0 \\ x_t \\ 0 \end{pmatrix} \leftarrow y_t$

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 et al. ICCV 2015]
"context-aware CNNs for person head detection"

instead : $\psi_t(x_t, y_t) = \left(NN_\theta(x_t)\right)^\top y_t$

II) "end-to-end" training : structured prediction energy network (SPEN)

III) recurrent neural networks (RNN)

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

graphical model approach

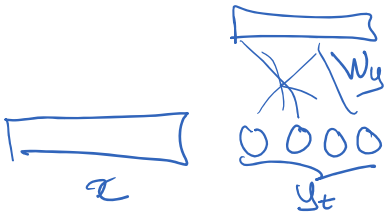
$\prod_t p(y_t | y_{\pi_t}, x)$ [directed]

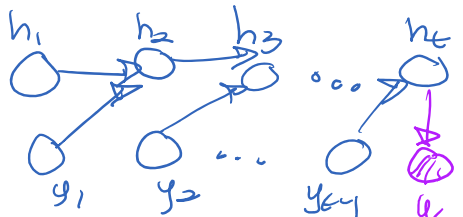
$\perp\!\!\!\perp$ $\frac{1}{Z(x)} \prod_c \psi_c(y_c; x)$ [undirected]

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

using a NN

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





$$h_t = f(f(f(\dots(y_0)\dots), x, y_{t-1}, W)$$

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

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

Standard learning : use maximum likelihood

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

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

"teacher's forcing"

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

Clar SGD on this

chain rule ⇝ backpropagation

ie. do not know $p(y \mid \text{unseen } y_t\text{'s}, x)$

14h40

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 $\hat{y}_t = \operatorname{argmax}_{y_t \in \mathcal{Y}_t} p(y \mid \hat{y}_{1:t-1}, x)$
- beam search "greedy decoding with memory of size k" (= "beam")

beam search: construct $y_1, \dots, y_T$

beam of size L (memory)

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

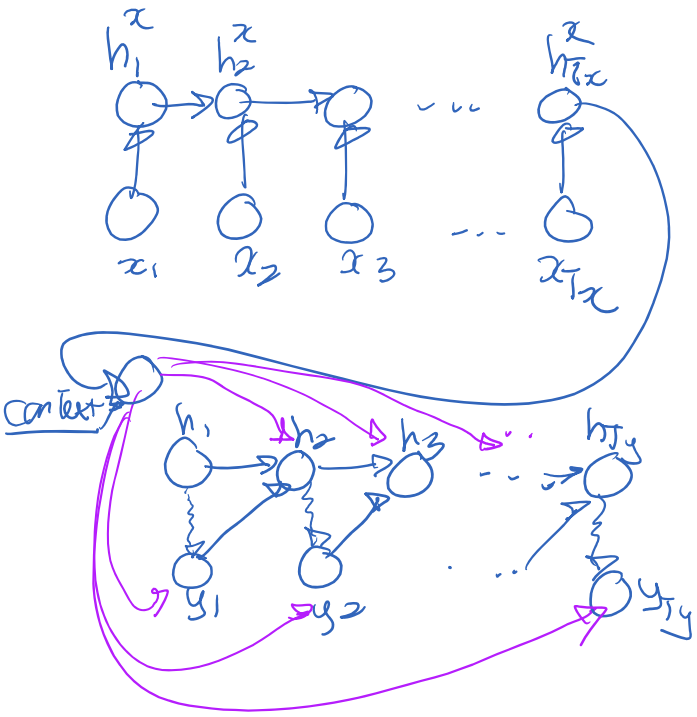
  then keep top L candidates as $\{y_{1:t+1}^{(\ell)}\}_{\ell=1}^{L}$

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

Seq2seq / encoder/architecture

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

when x can be variable length



"encoder RNN"

"decoder RNN"

issues:

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

b) long input sequence x?

problem: need to be summarized in fixed length context vector

solution: "attention mechanism"

c) vanishing gradient?

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