Lecture 2 - scribbles - EBM

Friday, January 26, 2018
13:35

today's - OCR example
- energy based models

OCR - optical character recognition example

x : sequence of images of characters 

$x = (x_1, \dots, x_{L_x})$ $\quad x_p \in \{0,1\}^{16 \times 8}$

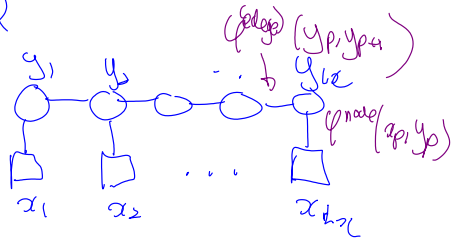
y : B R A C E

$\mathcal{Y}(x) = \Sigma^{L_x}$ $\quad \Sigma = \{A, B, \dots, Z\}$

in Max-Margin Markov network (M^3-net) paper:

$$\langle w, \varphi(x,y) \rangle = \sum_{p=1}^{L_x} \langle w^{(node)}, \varphi^{(node)}(x_p, y_p) \rangle + \sum_{p=1}^{L_x - 1} \langle w^{(edge)}, \varphi^{(edge)}(y_p, y_{p+1}) \rangle$$

graphical model



$$p(y|x) = \frac{1}{Z_w(x)} \exp\left(\langle w, \varphi(x,y) \rangle\right) = \frac{1}{Z_w(x)} \prod_{c \in \mathcal{C}} \psi_c(x, y_c)$$

here $C = \{p, p+1\}$ (edge)

$\Rightarrow$ can compute $\operatorname{argmax}_y \langle w, \varphi(x,y) \rangle$ using max-product alg. or viterbi alg. or max-sum

notation: $y_c \triangleq (y_i)_{i \in c}$

feature function:

node: $\varphi(x_p, y_p) = \begin{pmatrix} \text{vector}(x_p) \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ ← y_p^{th} position

16·8 · 26 (# of characters)

$\langle w, \varphi(x_p, y_p) \rangle = 0 + 0 + \ldots + \langle w_{y_p}, x_p \rangle + 0 \ldots$

w_{y_p} is a template for character y_p

w_a [a]

w_b [b]

edge feature:

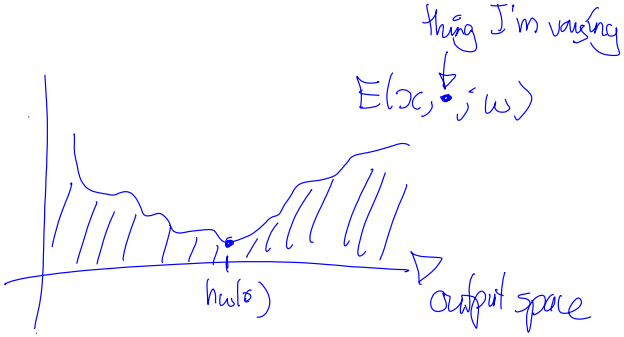
$\varphi(y_p, y_{p+1}) = \begin{pmatrix} \vdots \\ \mathbb{1}\{y_p = y_p, y_{p+1} = y_{p+1}\} \\ \vdots \end{pmatrix}$ ↓ (y_p, y_{p+1}) ↓ 26^2

$\langle w^{(edge)}, \varphi(y_p, y_{p+1}) \rangle = w^{(edge)}_{y_p, y_{p+1}}$

energy based methods: (LeCun et al. 2006)

model: $h_w(x) = \underset{y \in \mathcal{Y}(x)}{\text{argmin}}\ E(x, y; w)$ "energy fct."

$= \underset{y \in \mathcal{Y}(x)}{\text{argmax}}\ S(x, y; w)$ "score / compatibility fct."



ingredients:

modeling:
1) what is $E(x, y; w)$? e.g. $S(x, y; w) = \langle w, \varphi(x, y) \rangle$
2) how do you compute $\underset{y \in \mathcal{Y}(x)}{\text{argmin}}\ E(x, y; w)$? "inference" / "decoding"

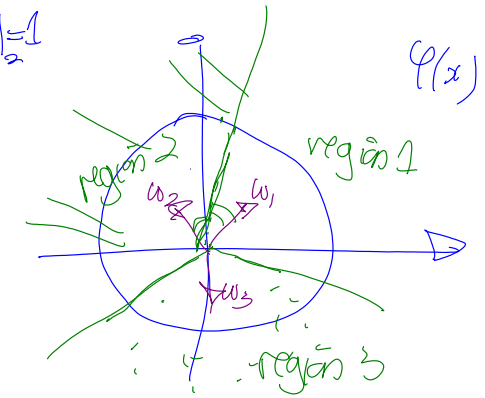
learning:
3) how to evaluate $E(x, y, w)$ on training set? → surrogate loss $\hat{\ell}(w)$
in general: $\tilde{\ell}(x^{(i)}, y^{(i)}, E(\cdot, \cdot, \cdot))$ "loss functional"
4) how to minimize $\hat{\ell}(w)$ to learn w? → optimization tricks

4) how to minimize $\hat{\ell}(w)$ to learn w? → optimization tricks

flat multiclass case:

"flat" setting $h_w(x) = \underset{y}{\operatorname{argmax}} \langle w_y, \varphi(x) \rangle$ $\varphi(x) \in \mathbb{R}^d$

$\|w_y\|_2 = 1$



equivalent to:

$\varphi(x,y) = (0, 0, \varphi(x), 0, 0) \in \mathbb{R}^{d \cdot K}$ (K = # of classes; $\varphi(x)$ in the y^{th} position)

surrogate losses:

$\hat{\ell}(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(x^{(i)}, y^{(i)}, w) + R(w)$

$h_w(x) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \langle w, \varphi(x,y) \rangle$

I) perceptron loss [Collins 2002 EMNLP]

$\ell(x, y, w) = \max_{\tilde{y} \in \mathcal{Y}(x)} S(x, \tilde{y}, w) - S(x, y, w)$ (score of ground truth)

$S(x, y, w) = \langle w, \varphi(x,y) \rangle$

$\max_{\tilde{y}} \langle w, \varphi(x,\tilde{y}) - \varphi(x,y) \rangle \geq 0$ by using $\tilde{y} = y$

observations: 1) degenerate solution $w = 0$ or constant score over y

does not converge in general

2) averaged perceptron algorithm:

- run constant stepsize stochastic subgradient method on $\hat{S}(w)$
- output $\hat{w}_T = \frac{1}{T+1}\sum_{t=0}^{T} w_t$ (Polyak averaging) → will converge to $w^* = 0$ (?)

comments: 1) Collins paper → he gives error bound and generalization error guarantees for perceptron

2) connection with the "herding" algorithm by Welling & al

"3rd way to learn" see ICML 2012

data —learn→ $\hat{w}_{ML}$

data —herding→ { sampling $y^{(t)}$

II) log-loss (probabilistic interpretation)

suppose $p(y|x;w) \propto \exp(\beta S(x,y;w))$

β: inverse temperature parameter

Boltzmann dist. in physics $\left(\beta = \frac{1}{k_B T}\right)$

MCL → log-loss:

$$\ell(x,y;w) = -\frac{1}{\beta}\log p(y|x;w) = \frac{1}{\beta}\log\left[\frac{\exp(\beta S(y))}{\sum_{\tilde{y}}\exp(\beta S(\tilde{y}))}\right]$$

($S(x,y;w)$; β: rescaling; $\sum_{\tilde{y}}\exp(\beta S(\tilde{y})) = Z_\beta(x;w)$ partition function)

$$= \frac{1}{\beta}\log\left(\sum_{\tilde{y}}\exp(\beta S(\tilde{y}))\right) - S(y)$$

NN / $\exp(S_y)$

$$= \frac{1}{\beta} \log\left(\sum_{\tilde{y}} \exp(\beta s(\tilde{y}))\right) - s(y)$$

$$\nabla N\left(\frac{\exp(s_y)}{\sum_{\tilde{y}} \exp(s_{\tilde{y}})}\right)_y$$

→ they call it soft-max
I call it soft-argmax

"log-sum-exp" ⟶ "soft-max" Let $\hat{y} = \operatorname{argmax}_{\tilde{y}} s(\tilde{y})$

why?

$$\frac{1}{\beta} \log\left[\exp(\beta s(\hat{y})) \left[\sum_{\tilde{y}} \exp\left(\beta \underbrace{(s(\tilde{y}) - s(\hat{y}))}_{\leq 0}\right)\right]\right]$$

$$\leq |\mathcal{Y}|$$

$$= s(\hat{y}) + \frac{1}{\beta} \log\left(\underbrace{\text{stuff}}_{\leq |\mathcal{Y}|}\right)$$

$\beta \to \infty$ (ie. zero-temperature limit) $\frac{1}{\beta} \log\left(\sum_{\tilde{y}} \exp(\beta s(\tilde{y}))\right) \xrightarrow{\beta \to \infty} \max_{\tilde{y}} s(\tilde{y})$

$\lim_{\beta \to \infty}$ log-loss(β) ⇝ perceptron loss?