Lecture 3 - Surrogate losses

Tuesday, January 24, 2023 2:49 PM

today: energy based methods & surrogate losses
multiclass

OCR - optical character recognition example

x : sequence of images of characters  $x = (x_1, \dots, x_{L_x})$

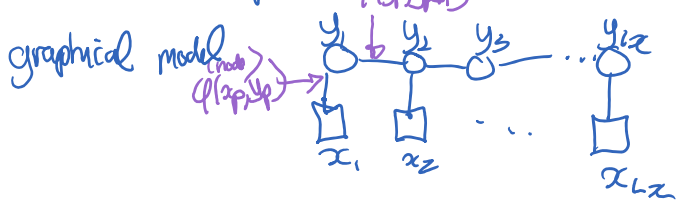
$x_p \in \{0,1\}^{16\times 8}$

y B R A C E $y(x) = \mathcal{E}^{L_x}$

$\mathcal{E} = \{A, \dots, Z\}$

in max-margin Markov network (M^3-net) papers:

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

graphical model 

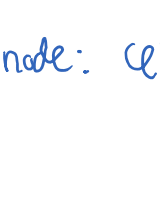
$$p_w(y|x) = \frac{1}{Z_w(x)} \exp(\langle w, \varphi(x,y)\rangle) = \frac{1}{Z_w(x)} \prod_{c\in\mathcal{C}} \psi_c(x_c, y_c)$$

where $C = \{p, p+1\}$

notation $y_c \triangleq (y_i)_{i\in C}$ ⇒ can compute $\arg\max_y \langle w, \varphi(x,y)\rangle$

using max product dg, aka. Viterbi dg or max sum

dimensions: $16 \cdot 8 \cdot 26$ (26 = # of characters)

node: $\varphi^{(node)}(x_p, y_p) =$  ← y_p^{th} position

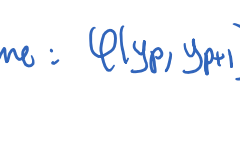
w_a 

w_b 

$\langle w, \varphi^{(node)}(x_p, y_p)\rangle$

$= 0 + 0 + \langle w_{y_p}, x_p\rangle + 0$

template for y_p

edge feature: $\varphi(y_p, y_{p+1}) =$  26^2

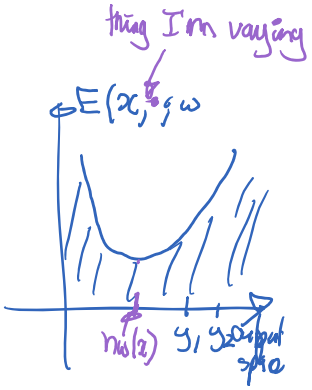
$\langle w^{(edge)}, \varphi(y_p, y_{p+1})\rangle = w^{(edge)}_{y_p, y_{p+1}}$

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

energy based methods : [Lecun & al 2006]

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

$= \underset{y \in \mathcal{Y}(x)}{\operatorname{argmax}} s(x, y; w)$ "score / compatibility"

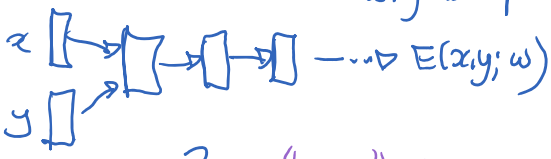


ingredients :

modeling

1) what is $E(x, y; w)$? e.g. $s(x, y; w) = \langle w, \varphi(x, y) \rangle$

or $E(x, y; w)$ scalar output of a NN with x & y as input



2) how do you compute (computational) $\underset{y \in \mathcal{Y}(x)}{\operatorname{argmin}} E(x, y; w)$? → "decoding" / "inference"

learning

3) how to evaluate $E(x, y; w)$ on a training set? → surrogate loss ("quality")

in general: $\mathcal{L}(x^{(i)}, y^{(i)}, E(\cdot, \cdot, w))$ → $\hat{\mathcal{L}}(w)$

"loss functional"

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

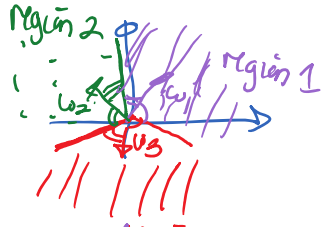
flat multiclass case

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

equivalent to $\varphi(x, y) = \begin{pmatrix} 0 \\ 0 \\ \varphi(x) \\ 0 \\ 0 \end{pmatrix}$ y^{th} position, $\in \mathbb{R}^{d \cdot K}$, K = # of classes

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

Visually: $\|w_y\| = 1$



region 3

Contrast this flat case with structured case
e.g. OCR : node feature map $\langle w, \varphi(x,y)\rangle = \sum_p \langle w, \varphi^{(node)}(x_p, y_p)\rangle$

→ have "sharing" of parameters between pieces of the joint labels
↝ "structure"

aside : in structured prediction, usually absorb "bias" in parameters $\tilde{\varphi}(x)$

$$\langle \tilde{w}, \tilde{\varphi}(x)\rangle = \langle w, \varphi(x)\rangle + b \qquad \begin{pmatrix}\varphi(x)\\ 1\end{pmatrix}$$

$$\tilde{w} = \begin{pmatrix} w \\ b \end{pmatrix}$$

open question: regularizing or not the bias
does it matter in structured pred. ?

15h45

Surrogate losses

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

I) perceptron loss [Collins et al 2002 EMNLP]

$$\ell^{percep}(x,y;w) = \left[\max_{\hat{y}\in\mathcal{Y}(x)} s(x,\hat{y};w) - s(x,y;w) \right]_+$$

($s(x,\hat{y};w)$: score of ground truth; $[\cdot]_+$: not needed if assume $y \in \mathcal{Y}(x)$ by using $\hat{y}=y$)

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

$$\max_{\tilde{y}} \langle w, \underbrace{\varphi(x,\tilde{y}) - \varphi(x,y)}_{-\Psi_x(\tilde{y})} \rangle \geq 0$$

observations: 1) degenerate solution to $\hat{\ell}(w)$ with $w=0$ or constant score over y

2) averaged perceptron alg. :

observations: 1) degenerate solution to $\hat{s}(w)$ with $w=0$ if constant zero over y

2) averaged perceptron alg.:

- amounts to running constant step size stochastic subgradient method on $\hat{s}(w)$
- output $\hat{w}_T = \frac{1}{T+1}\sum_{t=0}^{T} w_t$ (Polyak avg.)
  - ↳ will converge to $w^* = 0$ when data is not separable

comments: 1) Collin's paper → he gives error bound and generalization error guarantees for perceptron

2) (aside) connection with the "herding" alg. by Welling & al. [ICML 2012]

"3rd way to learn"

data $\xrightarrow{\text{learn}}$ $\hat{w}_{ML}$ $\xrightarrow{\text{sampling}}$ $y^{(t)}$

herding →

II) log-loss (CRF) (probabilistic interpretation)

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

↑ "inverse temperature" parameter

Boltzmann dist

$\beta = \frac{1}{k_B \cdot T}$ ← temperature

MCL → log-loss

$s(x,y;w)$

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

↑ rescaling; partition fct.

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

"log-sum-exp" → "soft max" why?

let $\hat{y} = \arg\max_{\tilde{y}} s(\tilde{y})$

let $\hat{y} = \underset{\tilde{y}}{\operatorname{argmax}}\, s(\tilde{y})$

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

($s(\tilde{y}) - s(\hat{y}) \leq 0$; the sum $\leq 1 + |\mathcal{Y}| - 1$)

$$= \frac{1}{\beta} s(\hat{y}) \cdot \beta + \frac{1}{\beta}\log(\text{stuff})$$

$\beta \to \infty$ (ie. zero temperature limit)

$$\frac{1}{\beta}\log\left(\sum_{y} \exp(\beta \cdot s(y))\right) \xrightarrow{\beta \to \infty} \max_{\tilde{y}} s(\tilde{y})$$

thus $\lim_{\beta \to \infty} \text{log-loss}(\beta) \longrightarrow$ perceptron loss

note:
- in deep learning they call "soft max"

$$\left(\frac{\exp(s(y))}{\sum_{\tilde{y}} \exp(s(\tilde{y}))}\right)_{y \in \mathcal{Y}}$$

I call this "soft-argmax"