Lecture 3 - surrogate losses

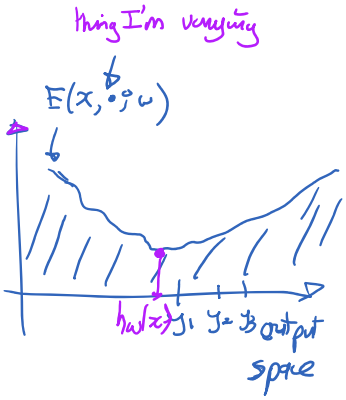
Tuesday, January 14, 2020 14:26

today: energy based methods & surrogate losses
multiclass

energy based methods : [LeCun et 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 $E(x, y; w)$ output of a NN with x & y as input

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

learning

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

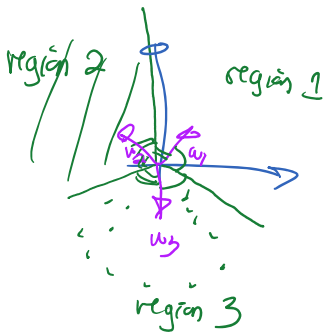
in general: $\hat{\ell}(x^{(i)}, y^{(i)}, E(\cdot, \cdot; w))$ "loss functional"

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

flat multiclass case:

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

visually: $\|w_y\| = 1$



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

contrast this flat case with structured case :

e.g. OCR node feature map $\langle w, \varphi^{(node)}(x, y) \rangle = \sum_p \langle w, \varphi^{(node)}(x_p, y_p) \rangle$

e.g. OCR node feature map $\langle w, \varphi^{(node)}(x,y)\rangle = \sum_p \langle w, \varphi^{(node)}(x_p, y_p)\rangle$

piece of sequence $= \sum_p \sum_{y_p'} \mathbb{1}\{y_p = y_p'\} \langle w_{y_p'}, \varphi(x)\rangle$

→ here "sharing" of parameters between different pieces of the joint labels → "structure"

aside: in structured prediction, usually absorb "bias" in parameters

standard binary classification $\operatorname{sgn}(\langle w, x\rangle + b)$

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

open question: regularizing or not the bias in structured prediction, does it matter?

Surrogate losses:

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

I) perceptron loss [Collins & al. 2002 EMNLP] score of ground truth

$$\ell(x, y; w) = \Big[\max_{\tilde{y} \in \mathcal{Y}(x)} s(x, \tilde{y}, w) - s(x, y, w)\Big]_+ \quad \text{(assume } y \in \mathcal{Y}(x)\text{)}$$

not regularized

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

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

by using $\tilde{y} = y$

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

2) averaged perceptron alg.: doesn't converge in general

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

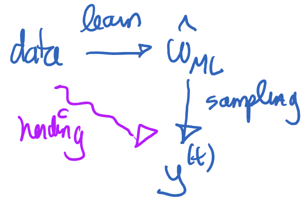
comments: 1) Collins' paper → he gives error bound and generalization error guarantees for perceptron

2) (aside) connection with the "hashing" alg. by Welling & al.

"3rd way to learn" [see ICML 2012]

.. learn ∧

→ way to learn [see ICML 2012]



15h36

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

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

β: "inverse temperature" parameter

Boltzmann dist. in physics

$\beta = \frac{1}{k_B T}$

MCL → log-loss

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

$\frac{-1}{\beta}$: rescaling; $s(y)$: $s(x,y;w)$; denominator: $Z_\beta(x;w)$ partition fct.

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

"log-sum-exp" ⇝ "softmax" why?

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

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

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

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

(stuff $\leq |\mathcal{Y}|$)

NOTE: in deep learning book, they call this "softmax"

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

I call this "soft-argmax"

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

thus $\lim_{\beta \to \infty}$ log-loss(β) → perceptron loss

III) Structured hinge loss

$$\ell(x,y;w) = \max_{\tilde{y}\in\mathcal{Y}(x)} \underbrace{[s(x,\tilde{y};w) + \ell(y,\tilde{y})]}_{\text{"loss-augmented decoding"}} - s(x,y;w)$$

a) Cartoon :

$\Rightarrow \ell^{sum}(x,y;w) = 0$

b) $\ell(x,y;w) \geq \ell(y, h_w(x))$

why? $\ell(x,y;w) = \max_{\tilde{y}} [s(\tilde{y}) + \ell(\tilde{y})] - s(y)$

$\geq s(\hat{y}) + \ell(\hat{y}) - s(y)$ let $\hat{y} = \operatorname{argmax}_{\tilde{y}\in\mathcal{Y}(x)} s(\tilde{y}) = h_w(x)$

if $y \in \mathcal{Y}(x) \Rightarrow s(\hat{y}) \geq s(y)$

$\geq \ell(\hat{y}) = \ell(y, \underbrace{h_w(x)}_{\hat{y}})$ //

binary case :

$y \in \{-1, +1\}$ $\quad w = \begin{pmatrix} w_+ \\ w_- \end{pmatrix}$ $\quad \ell(x,+1) = \begin{pmatrix} \ell(x) \\ 0 \end{pmatrix}$

$h_w(x) = \operatorname{argmax} \{\langle w_+, x\rangle, \langle w_-, x\rangle\}$

predict +1 if $\langle w_+, x\rangle \geq \langle w_-, x\rangle$

$\Leftrightarrow \langle w_+ - w_-, x\rangle \geq 0$

$\tilde{w} = w_+ - w_-$ $\quad h_w(x) = \operatorname{sgn}(\langle \tilde{w}, x\rangle)$

structured hinge loss

$$\ell^{sum}(x,y;w) = \max \{\langle w_+, x\rangle + \underbrace{\ell(y,+)}_{\mathbb{1}\{y\neq+1\}}, \langle w_-, x\rangle + \underbrace{\ell(y,-)}_{1-\mathbb{1}\{y\neq-1\}}\} - \langle w_y, x\rangle$$

$w_+ = \tilde{w} + w_-$

$= \max \{\langle \tilde{w}, x\rangle + \langle w_-, x\rangle + \mathbb{1}\{y\neq+1\}, \langle w_-, x\rangle + 1 - \mathbb{1}\{y\neq-1\}\} - \langle w_y, x\rangle$

$= \max \{\langle \tilde{w}, x\rangle + \mathbb{1}, 1 - \mathbb{1}\} + \langle w_-, x\rangle - \langle w_y, x\rangle$

case $y=+1$: $\max\{\langle \tilde{w}, x\rangle, 1\} - \langle \tilde{w}, x\rangle = [1 - \underbrace{y}_{+1}\langle \tilde{w}, x\rangle]_+$

case $y=-1$ $\max\{\langle \tilde{w}, x\rangle + 1, 0\} + 0 = [1 - y\langle \tilde{w}, x\rangle]_+$

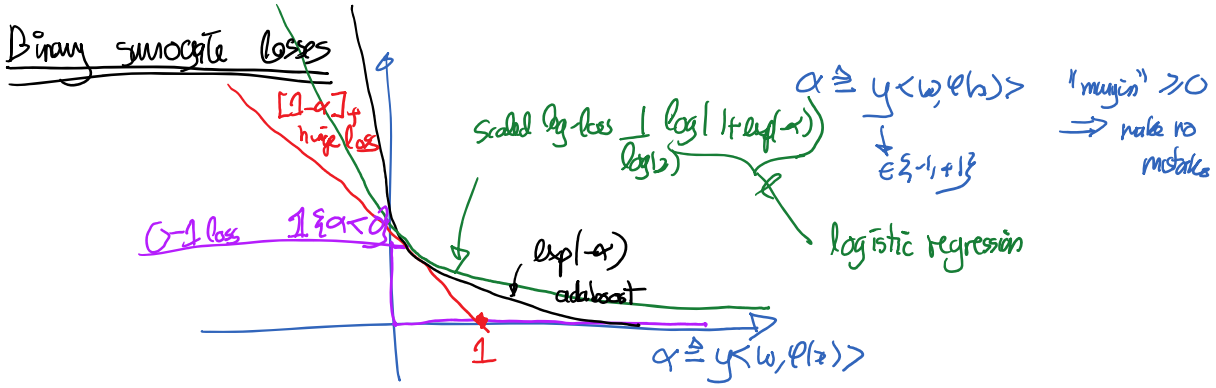
case $y=-1$ $\quad \max\{\langle \tilde{w}, x\rangle + 1, 0\} + 0 = [1 - y\langle \tilde{w}, x\rangle]_+$

overall: $[1 - y\langle \tilde{w}, x\rangle]_+$

structured SVM

$\ell(x, y; w) = [1 - y\langle \tilde{w}, x\rangle]_+$ ie. structured hinge loss reduces to binary SVM hinge loss when using $\ell(y, y') = \mathbb{1}\{y \neq y'\}$ and $\mathcal{Y} = \{-1, +1\}$

where $\tilde{w} = w_+ - w_-$



[Bartlett & al. 2006] $\rightarrow$ showed all these methods are consistent