Lecture 3 - Surrogate losses

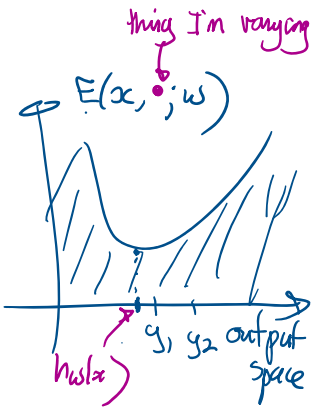
Tuesday, January 16, 2024 9:27 AM

today: · energy based methods & surrogate losses
· multiclass

energy based methods: [Lecun & al. 2006]

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

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

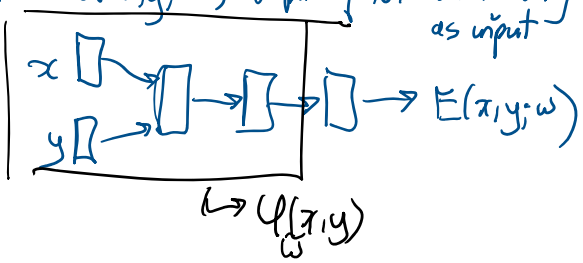


ingredients:

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 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 "quality" $E(x,y;w)$ on a training set? → surrogate loss $\hat{\mathcal{L}}(w)$

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

"loss functional"

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

flat multiclass case

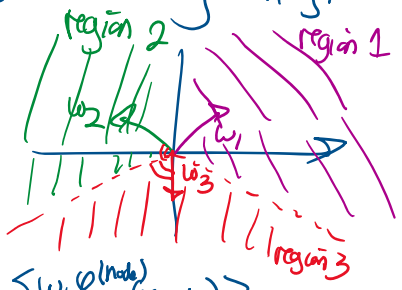
"flat" (ie. non-structured) setting $h_w(y) = \underset{y}{\operatorname{argmax}} \langle w_y, \varphi(x) \rangle$ $\varphi(x) \in \mathbb{R}^d$

equivalent to $\varphi(x,y) = \begin{pmatrix} 0 \\ 0 \\ \varphi(x) \\ \vdots \end{pmatrix} \in \mathbb{R}^{d \cdot k}$ with $w = \begin{pmatrix} \\ \\ \\ \end{pmatrix}$ (to simplify)

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

(to simplify picture)

visually: $\|w_y\| = 1$



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

contrast this flat case with structured case:

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

$\leq \sum_p \sum_{y'_p} \mathbb{I}\{y_p, y'_p\} \langle w_{y'_p}, \varphi(x) \rangle$

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

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

standard binary classification $\text{sgn}(\langle w, x \rangle + b)$ → $\tilde{\varphi}(x) = \begin{pmatrix} \varphi(x) \\ 1 \end{pmatrix}$

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

open question: regularizing or not the bias

does it matter in structured prediction?

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]

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

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

not needed if assume $y \in \mathcal{Y}(x)$

$\max_{\tilde{y}} \langle w, \underbrace{\varphi(x,\tilde{y}) - \varphi(x,y)}_{\triangleq -\psi(\tilde{y})} \rangle \geqslant 0$ by using $\tilde{y} = y$

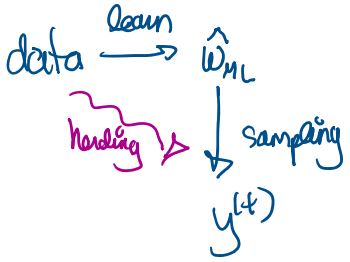
$\triangleq -\gamma(\tilde{y})$ by using $\tilde{y}=y$

observations: 1) degenerate solution to $\beta(w)$ with $w=0$ or constant score over y

2) averaged perceptron alg
   - does not converge in general
   - amounts to running constant step-size stochastic subgradient method on $\beta(w)$
   - output $\hat{w}_T = \frac{1}{T+1}\sum_{t=0}^{T} w_t$ (Polyak avg.) $(R(w)=0)$
     ↳ 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. Welling & al. [ICML 2012]
"3rd way to learn"



lon3)

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

suppose $p(\tilde{y}|x;w) \propto \exp(\beta s(x,\tilde{y};w))$
β "inverse temperature parameter"

Boltzmann dist.
$\beta = \frac{1}{k_B T}$ a temperature

MCL → log-loss

$$\ell(x,y;w) = -\frac{1}{\beta}\log p(y|x;w) = -\frac{1}{\beta}\log\left(\frac{\exp(\beta s(x,y;w))}{\sum_{\tilde{y}}\exp(\beta s(\tilde{y}))}\right) Z_\beta(x;w)$$
β rescaling; $s(x,y;w)$; $Z_\beta(x;w)$ "partition fct."

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

"log-sum-exp" → "soft max" why?

"log-sum-exp" → "soft max" why?

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

$$\frac{1}{\beta}\log\Big[\exp(\beta s(\hat{y}))\Big]\Big[\sum_{\tilde{y}} \exp\big(\beta(s(\tilde{y}) - s(\hat{y}))\big)\Big] \quad (\text{exponent} \le 0)$$

$$= \frac{\beta}{\beta} s(\hat{y}) + \frac{1}{\beta}\log(\text{stuff}) \quad (\text{stuff} \le |\mathcal{Y}|)$$

note: in deep learning book they call soft max

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

I call this "soft argmax" thus $\lim_{\beta \to \infty}$ log-loss(β) → perceptron loss

III) structured hinge loss

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

a) cartoon:

want: $s(y) \ge s(y_{\text{next}}) + \ell(y, y_{\text{next}})$ ✓

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

b) $\boxed{\ell^{\text{sum}}(x, y; w) \ge \ell(y, h_w(x))}$

why? $\ell(x, y, w) = \max_{\tilde{y}} [s(\tilde{y}) + \ell(\tilde{y})] - s(y)$ let $\hat{y} = \operatorname{argmax}_{\tilde{y}} s(\tilde{y}) = h_w(x)$

$\ge s(\hat{y}) + \ell(\hat{y}) - s(y)$

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

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

$$\frac{1}{\beta}\log\Big(\sum_{\tilde{y}} \exp\big(\beta[s(\tilde{y}) + \ell(\tilde{y})]\big)\Big) - s(y)$$

↳ hybrid between CRF loss & SVM loss

binary case: for structured hinge loss

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

$$h_w(x) = \arg\max \{ \langle w_+, \varphi(x) \rangle, \langle w_-, \varphi(x) \rangle \}$$

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

$$\Leftrightarrow \langle \underbrace{w_+ - w_-}_{\tilde{w} \triangleq w_+ - w_-}, \varphi(x) \rangle \geq 0$$

$$h_w(x) = \text{sgn}(\langle \tilde{w}, \varphi(x) \rangle)$$

Let's show $\ell^{SVM}(x,y;w) = [1 - y\langle \tilde{w}, x \rangle]_+$ where $\tilde{w} = w_+ - w_-$

(recopy boring derivation here) :)

i.e. structured hinge loss reduces to binary SVM hinge loss when using $\ell(y,y') = \mathbb{1}\{y \neq y'\}$ and $\mathcal{Y} = \{-1, +1\}$

(recopied from 2020 notes -- derivations for binary SVM:)

structured hinge loss

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

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

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

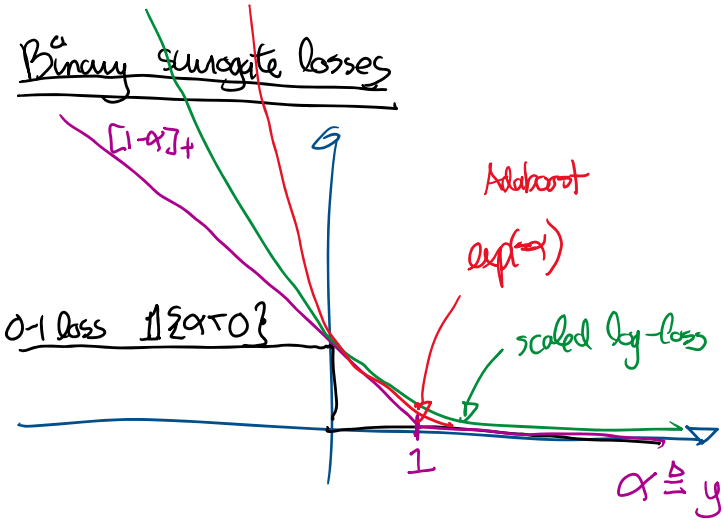
$$= \max \{ \langle \tilde{w}, x \rangle + \langle w_-, x \rangle + \mathbb{1}\{y \neq +1\}, \langle w_-, x \rangle + 1 - \mathbb{1}\{y \neq +1\} \} - \langle w_y, x \rangle$$

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

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

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



"margin"
$y\langle w, \phi(x)\rangle \geq 0 \Rightarrow$ make no mistake

[Bartlett et al. 2006] → showed all these methods are consistent