Lecture 14

Logistic regression

Discriminative classification model $p(y|\vec{x},\vec{\theta})$, where $y \in \{1,\dots,C\}$ is the class label, $\vec{x} \in R^D$ is the input vector, and $\vec{\theta}$ are model prms. (C = # classes; $\vec{\theta} = (\vec{w}, b)$)

Recall that $p(y=1|\vec{x},\vec{\theta}) = \sigma(a) = \frac{1}{1+e^{-a}}$,

where $a = \vec{w}\cdot\vec{x} + b$.

$$p(y=0|\vec{x},\vec{\theta}) = \sigma(-a).$$

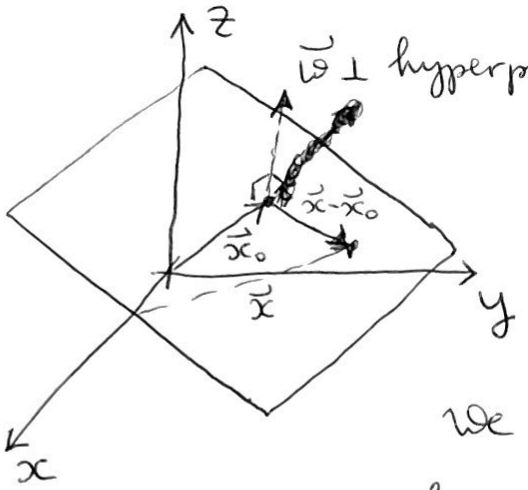
Indeed, $\sigma(a) + \sigma(-a) = \frac{1}{1+e^{-a}} + \frac{1}{1+e^{a}} =$

$$= \frac{1+e^{a}+1+e^{-a}}{(1+e^{-a})(1+e^{a})} = 1. \quad =$$

If we define $f(\vec{x},\vec{\theta}) = b + \vec{w}\cdot\vec{x} = b + \sum_{i=1}^{D} w_i x_i$,

$f(\vec{x},\vec{\theta}) = 0$ is the decision boundary - a hyperplane in $\vec{x}$-space.

Consider a plane defined by a normal vector $\vec{w}$, and going through some point $\vec{x}_0$.



For $\forall \vec{x} \in$ hyperplane,

$\vec{w}\cdot(\underbrace{\vec{x}-\vec{x}_0}_{\text{lies in the hyperplane}}) = 0$.

We are free to choose

$b = -\vec{w}\cdot\vec{x}_0 \Rightarrow \underbrace{\vec{w}\cdot\vec{x}+b=0}_{\text{decision boundary (DB)}}$

Non-linear input features:

$\vec{x} \Rightarrow \vec{\varphi}(x)$

$\vec{x}$: D; $\vec{\varphi}(x)$: $D' \neq D$ in general

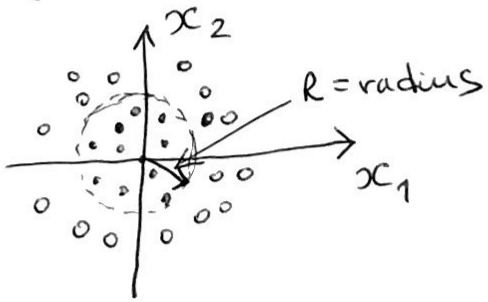
This can create non-linear decision boundaries.

Ex. $\vec{\varphi}(x_1, x_2) = (1, x_1^2, x_2^2)$ [$b=0$ here]

Using $\vec{w} = (-R^2, 1, 1)$ we obtain

$\vec{w}\cdot\vec{\varphi}(\vec{x}) = x_1^2 + x_2^2 - R^2 = 0$ @ DB

DB eq'n is a circle: $x_1^2 + x_2^2 = R^2$



Objective function

$$\underbrace{-\mathcal{L}(\vec{w})}_{\text{negative log } \mathcal{L} \text{ (NLL)}} = -\frac{1}{N}\sum_{n=1}^{N} \log\left[\mu_n^{y_n}(1-\mu_n)^{1-y_n}\right] =$$

$y_n = \{0, 1\}$

$\mu_n = \sigma(a_n)$, where $a_n = \vec{w}\cdot\vec{x}_n + b =$

$$= \sum_{i=1}^{D+1} w_i \cdot x_{n,i} \quad (\uparrow)$$

$w_{D+1} = b$, $x_{n,D+1} = 1$ $(\forall n)$

$$= -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log\mu_n + (1-y_n)\log(1-\mu_n)\right] =$$

$$= \frac{1}{N}\sum_{n=1}^{N} H(y_n, \mu_n), \text{ where}$$

$H(p, q) = -p\log q - (1-p)\log(1-q)$ is the binary cross-entropy

Optimization

$$-\frac{\partial}{\partial w_i}\mathcal{L}(\vec{w}) = g_i(\vec{w}) \qquad i = 1, \ldots, D+1$$

Note that $\frac{\partial}{\partial w_i}\underbrace{\mu_n}_{\sigma(\underbrace{\vec{w}\cdot\vec{x}_n}_{a_n})} = \mu_n(1-\mu_n)\frac{\partial a_n}{\partial w_i} =$

$$= \mu_n(1-\mu_n)\, x_{n,i}.$$

Then $\frac{\partial}{\partial w_i}(-\mathcal{L}(\vec{w})) = -\frac{1}{N}\sum_{n=1}^{N}\left[\frac{y_n}{\mu_n}\mu_n(1-\mu_n)x_{n,i} + \right.$

$\left. + \frac{1-y_n}{1-\mu_n}(-1)\mu_n(1-\mu_n)x_{n,i}\right] =$

$= \frac{1}{N}\sum_n \left[(1-y_n)\mu_n - y_n(1-\mu_n)\right]x_{n,i} =$

$= \frac{1}{N}\sum_n \underbrace{[\mu_n - y_n]}_{\text{'error signal'}} x_{n,i}$.

Next, we compute the Hessian.

$$\frac{\partial^2}{\partial w_i \partial w_j}(-\mathcal{L}(\vec{w})) = \frac{1}{N}\frac{\partial}{\partial w_j}\left(\sum_n [\mu_n - y_n]x_{n,i}\right) =$$

$$= \frac{1}{N}\sum_n \mu_n(1-\mu_n)x_{n,i}x_{n,j} .$$

If we define ($\leftarrow$ pos. #s on the diagonal)

$$S_{N\times N} = \text{diag}[\mu_1(1-\mu_1), \ldots, \mu_N(1-\mu_N)],$$

we obtain

$$H(\vec{w}) = \frac{1}{N} X^T S X \quad , \text{ where}$$

$$X_{N\times(D+1)} = \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_N \end{pmatrix}$$

Now, for $\forall\ \vec{v} \neq \vec{0}$,

$$\vec{v}^T H \vec{v} = v^T(X^T S X)v = (v^T X^T S^{1/2})\cdot(\underbrace{S^{1/2} X v}_{\vec{c},\ N\text{-dim vector}}) =$$

$$= \vec{c}^T \cdot \vec{c} > 0 .$$

Thus, $-\mathcal{L}(\vec{w})$ is strictly convex $\Rightarrow$

$\Rightarrow g(\vec{w})=0$ corresponds to the global min.

One can use SGD, for example:

$$\vec{w}_{t+1} = \vec{w}_t - \eta_t \underbrace{(\mu_{n_t} - y_{n_t})\vec{x}_n}_{|B|=1;\ n_t \text{ is the datapoint chosen @ } t}$$

One can also use Newton's method:

$$\vec{w}_{t+1} = \vec{w}_t - \underbrace{\eta_t}_{=1 \text{ for simplicity}} H_t^{-1} \vec{g}_t =$$

$$= \vec{w}_t + (X^T S_t X)^{-1} X^T (\vec{y} - \vec{\mu}_t) =$$

$$= (X^T S_t X)^{-1} \left[(X^T S_t X)\vec{w}_t + X^T(\vec{y} - \vec{\mu}_t) \right] =$$

$$= (X^T S_t X)^{-1} (X^T S_t) \left[\underbrace{X\vec{w}_t + S_t^{-1}(\vec{y} - \vec{\mu}_t)}_{\equiv \vec{z}_t \quad N\text{-dim}} \right]$$

Indeed, $z_{t,n} = \vec{w}_t \cdot \vec{x}_n + \dfrac{y_n - \mu_{t,n}}{\mu_{t,n}(1-\mu_{t,n})}$.

$n = 1, \ldots, N$

MAP estimation (to prevent overfitting)

Minimize $-\mathcal{L}(\vec{w}) + \lambda |\vec{w}|^2 \equiv \tilde{\mathcal{L}}(\vec{w})$

prior $p(\vec{w}) = \mathcal{N}(\vec{w} | \vec{0}, \lambda^{-1}\mathbb{I})$ $\{\lambda > 0$

Then $\begin{cases} \vec{\nabla}_{\vec{w}} \tilde{\mathcal{I}}(\vec{w}) = \vec{g}(\vec{w}) + 2\lambda \vec{w}, \\ \nabla^2_{\vec{w}} \tilde{\mathcal{I}}(\vec{w}) = H(\vec{w}) + 2\lambda \mathbb{I}. \end{cases}$

$\nabla^2_{\vec{w}}$: $\frac{\partial^2}{\partial w_i \partial w_j}$

Minimization can then be done as before

Standardization

The $\lambda |\vec{w}|^2$ penalty assumes that all w_i's are similar in magnitude. Thus, it is common to standardize the data:

$$x_{n,i} \to \tilde{x}_{n,i} = \frac{x_{n,i} - \hat{\mu}_i}{\hat{\sigma}_i}, \text{ where}$$

$$\begin{cases} \hat{\mu}_i = \frac{1}{N} \sum_{n=1}^{N} x_{n,i}, \\ \hat{\sigma}_i^2 = \frac{1}{N} \sum_{n=1}^{N} (x_{n,i} - \hat{\mu}_i)^2. \end{cases}$$