

Lecture 10

Logistic regression

Discriminative classification model

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

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

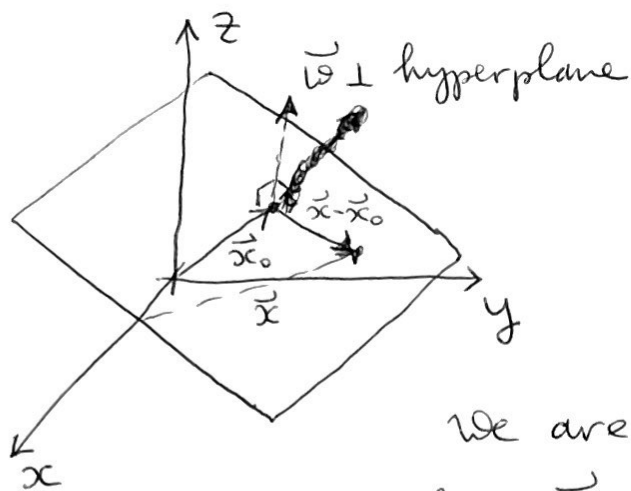
where $a = \vec{w} \cdot \vec{x} + b$.

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

$$\begin{aligned} \text{Indeed, } \sigma(a) + \sigma(-a) &= \frac{1}{1+e^{-a}} + \frac{1}{1+e^a} = \\ &= \frac{1+e^a + 1+e^{-a}}{(1+e^{-a})(1+e^a)} = 1. \end{aligned}$$

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

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:

$$\underbrace{\vec{x}}_D \Rightarrow \underbrace{\vec{g}(x)}_{D' \neq D \text{ in general}}$$

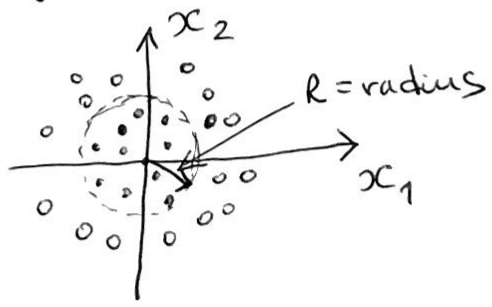
This can create non-linear decision boundaries.

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

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

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

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



Objective function

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

$$y_n = \{0, 1\}$$

$$\mu_n = \sigma(a_n), \text{ where } a_n = \vec{w} \cdot \vec{x}_n + b = \sum_{i=1}^{D+1} w_i \cdot x_{n,i}$$

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

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

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

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

Optimization

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

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

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

$$\begin{aligned}
\text{Then } \frac{\partial}{\partial w_i} (-\mathcal{L}(\vec{w})) &= -\frac{1}{N} \sum_{n=1}^N \left[\frac{y_n}{\mu_n} \mu_n (1-\mu_n) x_{n,i} + \right. \\
&\quad \left. + \frac{1-y_n}{1-\mu_n} (-1) \mu_n (1-\mu_n) x_{n,i} \right] = \\
&= \frac{1}{N} \sum_n \left[(1-y_n) \mu_n - y_n (1-\mu_n) \right] x_{n,i} = \\
&= \frac{1}{N} \sum_n \underbrace{[\mu_n - y_n]}_{\text{'error signal'}} x_{n,i} .
\end{aligned}$$

Next, we compute the Hessian.

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

If we define

pos. #s on the diagonal

$$S' = \text{diag}[\mu_1(1-\mu_1), \dots, \mu_N(1-\mu_N)],$$

we obtain

$$\begin{aligned}
H(\vec{w}) &= \frac{1}{N} X^T S' X, \quad \text{where} \\
X_{N \times (D+1)} &= \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_N \end{pmatrix}
\end{aligned}$$

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

$$\begin{aligned}
\vec{v}^T H \vec{v} &= \vec{v}^T (X^T S' X) \vec{v} = (\vec{v}^T X^T S'^{1/2}) \cdot (\underbrace{S'^{1/2} X \vec{v}}_{\vec{c}, \text{ N-dim vector}}) = \\
&= \vec{c}^T \cdot \vec{c} > 0 .
\end{aligned}$$

Thus, $-\mathcal{I}(\vec{w})$ is strictly concave $\Rightarrow$
 $\Rightarrow g(\vec{w})=0$ corresponds to the
 global min.

One can use SGD, for example:

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

One can also use Newton's method:
 $\stackrel{=1 \text{ for simplicity}}{\eta_t}$

$$\begin{aligned} \vec{w}_{t+1} &= \vec{w}_t - \eta_t H_t^{-1} \vec{g}_t = \\ &= \vec{w}_t + (X^T S_t' X)^{-1} X^T (\vec{y} - \vec{\mu}_t) = \\ &= (X^T S_t' X)^{-1} [(X^T S_t' X) \vec{w}_t + X^T (\vec{y} - \vec{\mu}_t)] = \\ &= (X^T S_t' X)^{-1} (X^T S_t) \underbrace{[X \vec{w}_t + S_t^{-1} (\vec{y} - \vec{\mu}_t)]}_{\vec{z}_t \text{ } N\text{-dim}} \end{aligned}$$

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

 $n=1, \dots, N$

MAP estimation (to prevent overfitting)

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

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



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

$\underbrace{\quad}_{\frac{\partial^2}{\partial w_i \partial w_j}}$

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:

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

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