Bayesian logistic regression

{Lecture 10

(K=2) here

Assume $p(\vec{w}) = \mathcal{N}(\vec{w} \mid \vec{0}, \underbrace{\alpha^{-1}\mathbb{I}}_{"S_0"})$

$$p(\vec{w}\mid\vec{t}) \sim p(\vec{t}\mid\vec{w})\, p(\vec{w})$$

Then $$\log p(\vec{w}\mid\vec{t}) = -\frac{\alpha}{2}\vec{w}^T\vec{w} + \sum_{n=1}^{N}\left[t_n \log y_n + (1-t_n)\log(1-y_n)\right] + \text{const}(\vec{w}), \quad (*)$$

where $y_n = \sigma(\vec{w}^T\cdot\vec{\varphi}_n)$ $\quad t_n = \{0,1\}$

(*) is at max when $\vec{w} = \vec{w}_{MAP}$; then

$$S_N \equiv -\vec{\nabla}_{\vec{w}}\vec{\nabla}_{\vec{w}}\log p(\vec{w}\mid\vec{t}) = \alpha\mathbb{I} + \sum_{n=1}^{N} y_n(1-y_n)\vec{\varphi}_n\vec{\varphi}_n^T$$

Thus $p(\vec{w}\mid\vec{t})$ (posterior) $\Rightarrow q(\vec{w}\mid\vec{t}) = \mathcal{N}(\vec{w}\mid\vec{w}_{MAP}, S_N)$.

Next, predictive distribution:

$$p(C_1\mid\vec{\varphi},\vec{t}) = \int d\vec{w}\, p(C_1\mid\vec{\varphi},\vec{w})\, p(\vec{w}\mid\vec{t}) \simeq \int d\vec{w}\, \sigma(\underbrace{\vec{w}^T\vec{\varphi}}_{"a"})\, q(\vec{w}\mid\vec{t})$$

($\vec{\varphi}$: new input vector of features)

$$p(C_2\mid\vec{\varphi},\vec{t}) = 1 - p(C_1\mid\vec{\varphi},\vec{t}).$$

Use $\sigma(\vec{w}^T\vec{\varphi}) = \int da\, \sigma(a)\,\delta(a-\vec{w}^T\vec{\varphi})$ to obtain:

$$p(C_1\mid\vec{\varphi},\vec{t}) \overset{\text{since } p(\vec{w}\mid\vec{t}) \text{ replaced by } q(\vec{w}\mid\vec{t})}{\simeq} \int da\,\sigma(a)\,p(a), \text{ where}$$

$$p(a) = \int d\vec{w}\, q(\vec{w}\mid\vec{t})\,\delta(a-\vec{w}^T\vec{\varphi})$$

So, compute

$$p(a) = \frac{1}{(2\pi)^{M/2}} \frac{1}{|S_N|^{1/2}} \int d\vec{\omega} \frac{1}{2\pi} \int dk\, e^{ik(a - \vec{\omega}^T \vec{g})} \times$$

$$\times e^{-\frac{1}{2}(\vec{\omega} - \vec{\omega}_{MAP})^T S_N^{-1} (\vec{\omega} - \vec{\omega}_{MAP})} \quad (=)$$

$$\uparrow \vec{\omega}' = \vec{\omega} - \vec{\omega}_{MAP}$$

$$(=) \frac{1}{(2\pi)^{M/2}} \frac{1}{|S_N|^{1/2}} \frac{1}{2\pi} \int dk\, e^{ik(a - \vec{g}^T \vec{\omega}_{MAP})} \times$$

$$\times \int d\vec{\omega}'\, e^{-ik \vec{g}^T \vec{\omega}'} e^{-\frac{1}{2} \vec{\omega}'^T S_N^{-1} \vec{\omega}'} \quad [=]$$

[If M is symm. & invertible,

$$\vec{x}^{\dagger} M \vec{x} - 2\vec{b}^{\dagger} \vec{x} = (\vec{x} - M^{-1}\vec{b})^{\dagger} M (\vec{x} - M^{-1}\vec{b}) - \vec{b}^{\dagger} M^{-1} \vec{b}$$

($\dagger$: Hermitian conjugate)]

where,

$$\begin{cases} \vec{x} \to \vec{\omega}', \\ M \to \frac{S_N^{-1}}{2} \Rightarrow M^{-1} = 2S_N, \\ 2\vec{b}^{\dagger} \to ik \vec{g}^T \Rightarrow \vec{b} \to -\frac{ik}{2} \vec{g}. \end{cases}$$

$$[=] \frac{1}{(2\pi)^{M/2}} \frac{1}{|S_N|^{1/2}} \frac{1}{2\pi} \int dk\, e^{ik(a - \vec{g}^T \vec{\omega}_{MAP})} e^{-\frac{k^2}{4} \vec{g}^T (2S_N) \vec{g}} \times$$

$$\times \underbrace{\int d\vec{\omega}'\, e^{-(\vec{\omega}' + ikS_N\vec{g})^{\dagger} \frac{S_N^{-1}}{2} (\vec{\omega}' + ikS_N\vec{g})}}_{(2\pi)^{\frac{M}{2}} |S_N|^{1/2}} =$$

$$= \frac{1}{2\pi} \int dk\, e^{ik(a - \vec{g}^T \vec{\omega}_{MAP})} e^{-\frac{k^2}{2} \vec{g}^T S_N \vec{g}}$$

Using $\int dx\, e^{ax^2+bx} = \sqrt{-\frac{\pi}{a}}\, e^{-b^2/4a}$,

we obtain:

$$\begin{cases} x \to k, \\ a \to -\frac{1}{2}\vec{\varphi}^T S_N \vec{\varphi}, \\ b \to i(a - \vec{\varphi}^T \vec{w}_{MAP}), \end{cases} \quad \text{so that}$$

$$p(a) = \underbrace{\frac{1}{2\pi}\sqrt{\frac{2\pi}{\vec{\varphi}^T S_N \vec{\varphi}}}}_{\frac{1}{\sqrt{2\pi\sigma_a^2}}} e^{-\frac{(a - \vec{\varphi}^T \vec{w}_{MAP})^2}{2\underbrace{\vec{\varphi}^T S_N \vec{\varphi}}_{\sigma_a^2}}} = \mathcal{N}(a | \underbrace{\vec{w}_{MAP}^T \vec{\varphi}}_{\mu_a}, \sigma_a^2).$$

Thus

$$p(C_1 | \vec{\varphi}, \vec{t}) = \int da\, \underbrace{\sigma(a)}_{\text{use } \tilde{\varphi}(\lambda a)} \mathcal{N}(a | \mu_a, \sigma_a^2) \, x =$$

$$= \varphi\left(\frac{\mu_a}{(\lambda^2 + \sigma_a^2)^{1/2}}\right) \approx \sigma\left(\frac{\mu_a}{(1 + \underbrace{\lambda^2}_{\frac{\pi}{8}}\sigma_a^2)^{1/2}}\right).$$

↑ shown below

Note that the DB ($p(C_1|\vec{\varphi}, \vec{t}) = p(C_2|\vec{\varphi}, \vec{t}) = 0.5$) is given by $\mu_a = 0$.

$\mu_a = \vec{w}_{MAP}^T \vec{\varphi}$, linear in feature space

Now, let's show that

$$\int da\, \Phi(\lambda a)\, \mathcal{N}(a|\mu_0, \sigma_0^2) = \Phi\left(\frac{\mu}{\sqrt{\frac{1}{\lambda^2}+\sigma^2}}\right)$$

RHS:

$$\Phi\left(\underbrace{\frac{\mu}{\sqrt{\frac{1}{\lambda^2}+\sigma^2}}}_{=z}\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} d\theta\, e^{-\theta^2/2}, \text{ s.t.}$$

$$\frac{d\Phi}{d\mu} = \frac{d\Phi}{dz}\frac{dz}{d\mu} = \frac{1}{\sqrt{2\pi}} e^{-\frac{\mu^2}{2\left(\frac{1}{\lambda^2}+\sigma^2\right)}} \frac{1}{\sqrt{\frac{1}{\lambda^2}+\sigma^2}}$$

LHS:

use $\frac{a-\mu}{\sigma} = z \Rightarrow a = \mu + \sigma z$

$da = \sigma dz$

$$\frac{1}{\sqrt{2\pi\sigma^2}}\, \sigma \int_{-\infty}^{\infty} dz\, \underbrace{\Phi(\lambda(\mu+\sigma z))}_{\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\lambda(\mu+\sigma z)} d\theta\, e^{-\theta^2/2}} e^{-z^2/2}$$

Then $\frac{d}{d\mu}(\text{LHS}) = \frac{1}{2\pi}\int_{-\infty}^{\infty} dz\, e^{-\frac{\lambda^2(\mu+\sigma z)^2}{2}} \lambda\, e^{-z^2/2} =$

$$= \frac{\lambda}{2\pi} e^{-\frac{\lambda^2\mu^2}{2}} \int dz\, e^{-\lambda^2\mu\sigma z} e^{-\frac{z^2}{2}(1+\lambda^2\sigma^2)} \quad (=)$$

Here, $\begin{cases} a = -\frac{1}{2}(1+\lambda^2\sigma^2) \\ b = -\lambda^2\mu\sigma \end{cases}$

$$(=)\ \frac{\lambda}{2\pi} e^{-\frac{\lambda^2\mu^2}{2}} \sqrt{\frac{2\pi}{1+\lambda^2\sigma^2}}\, e^{\frac{\lambda^4\mu^2\sigma^2}{2(1+\lambda^2\sigma^2)}} \quad (\Diamond)$$

$$\Leftrightarrow \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{\lambda^{-2}+\sigma^2}} e^{-\underbrace{\frac{\lambda^2\mu^2}{2}\left[1-\frac{\lambda^2\sigma^2}{1+\lambda^2\sigma^2}\right]}}$$

$$\frac{\lambda^2\mu^2}{2}\frac{1}{1+\lambda^2\sigma^2} = \frac{\mu^2}{2}\frac{1}{\lambda^{-2}+\sigma^2}$$

So, $\frac{d}{d\mu}(LHS) = \frac{d}{d\mu}(RHS) \Rightarrow LHS = RHS + const(\mu)$

Consider the $\mu \to +\infty$ limit:

$$\lim_{\mu\to\infty} RHS = 1$$

$$\lim_{\mu\to\infty} LHS = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dz\, e^{-\frac{z^2}{2}} = 1$$

$$\Phi(\lambda(\mu+\sigma z)) \to 1 \quad \text{as} \quad \mu \to +\infty$$

Thus $const(\mu) = 0 \Rightarrow LHS = RHS$

Neural networks

Idea: make basis functions weight-dependent and fit those weights.

Previously, we considered

$$y(\vec{x}, \vec{w}) = f\left(\sum_{j=1}^{M} w_j \varphi_j(\vec{x})\right), \text{ where}$$

$f(\cdot)$ is identity for regression and non-linear activation f'n for classification.

Now, consider $(x_1, \ldots, x_D)$ ← input vector

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \qquad j = 1, \ldots, M$$

(a_j: activation; $w_{ji}^{(1)}$: weights; $w_{j0}^{(1)}$: bias)

Then, $z_j = h(a_j)$ ← non-linear activation f'n

$h(\cdot)$ can be $\sigma(\cdot)$ or $\tanh(\cdot)$

Next, $a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}$ $\qquad k = 1, \ldots, K$ ↗ total # outputs

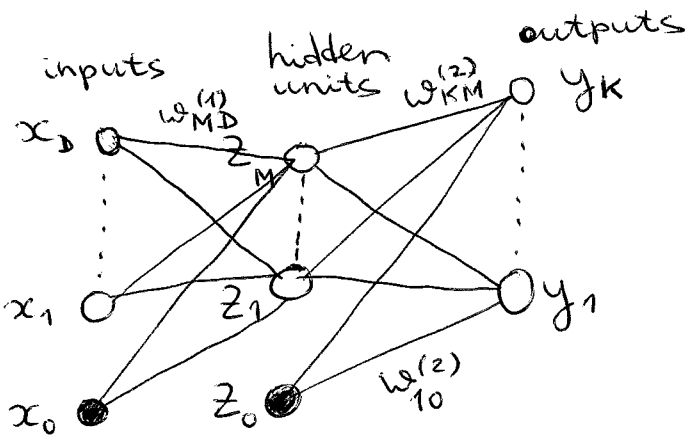
Finally, $y_k = \tilde{h}(a_k)$, where $\tilde{h}$ may be ($K=1$ or $K>1$) identity for regression, $\sigma(\cdot)$ for binary classification ($K=2$), etc.

(y_k: output vector)

We have:

$$y_k(\vec{x}, \vec{w}) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} \underbrace{h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \right)}_{z_j,\ j=1,\dots,M} + w_{k0}^{(2)} \right)$$

Note that it does not make sense to have $h(\cdot)$ as identity, since then the argument of the σ-function is just a linear model with various products of weights.



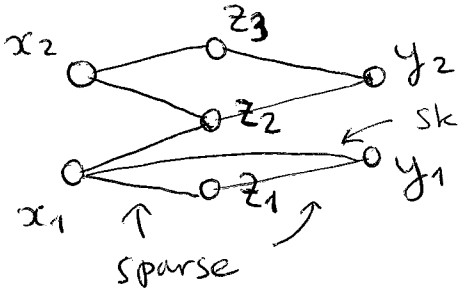
Define $x_0 = 1$ & $z_0 = 1$, then

$$y_k(\vec{x}, \vec{w}) = \sigma\left(\sum_{j=0}^{M} w_{kj}^{(2)} \underbrace{z_j}_{= \begin{cases} 1, & j=0 \\ h\left(\sum_{i=0}^{D} w_{ji}^{(1)} x_i \right), & j=1,\dots,M \end{cases}} \right)$$

<u>Generalizations</u>: (1) multiple layers of hidden units

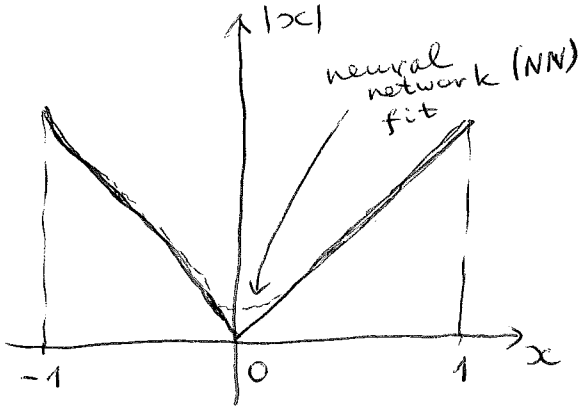
(2) sparse network architectures

(3) skip-layer connections:

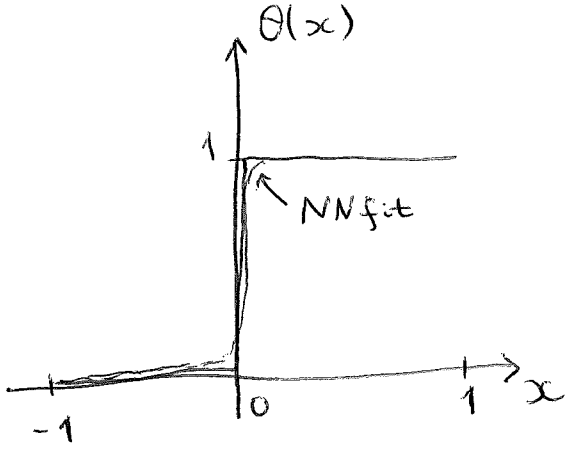


Note: feed-forward architectures <u>only</u>

Performance: can fit various functions fairly accurately



Sample $N = 50$ datapoints uniformly in $[-1, 1]$ interval, fit a two-layer network (input layer + hidden layer) discussed above:
3 hidden units,
tanh(·) activation f'n,
linear output units.



Weight-space symmetries:

with tanh(·) activation f's,

$$\tanh(-a) = -\tanh(a)$$, and

changing the sign of all weights (and the bias) leading into a unit can be compensated by the change in sign of all weights leading out of that unit.

M hidden units → 2^M equivalent weight vectors.

Similarly, can exchange weight values leading in and out of a hidden unit with another hidden unit ⇒ M! permutations

So we have $2^M M!$ symmetries for a two-layer network (can be easily generalized to other architectures and activation functions)