ML solution

{Lecture 8}

Recall that we want to use $\mathcal{D} = \{\vec{x}_n, \vec{t}_n\}$ dataset $n = 1, \dots, N$

$$p(C_i \mid \vec{x}, \mathcal{D}) = \frac{p(\vec{x} \mid C_i, \mathcal{D})\, p(C_i \mid \mathcal{D})}{\sum_{j=1}^{K} p(\vec{x} \mid C_j, \mathcal{D})\, p(C_j \mid \mathcal{D})}$$

Before we assumed that $p(\vec{x} \mid C_i)$ is a known gaussian and that $p(C_i)$ is also given. now let's estimate them by ML.

So formally speaking they do not depend on $\mathcal{D}$

Specifically, consider $\underline{K=2}$ for simplicity.

Define $\begin{cases} p(C_1 \mid \mathcal{D}) = \pi \quad \Rightarrow \quad p(C_2 \mid \mathcal{D}) = 1 - \pi; \\ t_n = 1 \text{ in } C_1, \quad t_n = 0 \text{ in } C_2. \end{cases}$

Then $\mathcal{J} = \prod_{n=1} \pi^{t_n} (1-\pi)^{1-t_n}$, or

$$\log \mathcal{J} = \sum_{n=1}^{N} \left\{ t_n \log \pi + (1 - t_n) \log(1-\pi) \right\}.$$

$$\frac{\partial}{\partial \pi} \log \mathcal{J} = \frac{1}{\pi} \sum_{n=1}^{N} t_n - \frac{1}{1-\pi} \sum_{n=1}^{N} (1 - t_n) = 0, \text{ or}$$

$$(1-\pi) \sum_{n=1}^{N} t_n - \pi \sum_{n=1}^{N} (1 - t_n) = 0,$$

$$\pi N = \sum_{n=1}^{N} t_n \Rightarrow \underset{p(C_1 \mid \mathcal{D})_{ML}}{\underset{\shortparallel}{\pi}} = \frac{1}{N} \underbrace{\sum_{n=1}^{N} t_n}_{N_1}$$

Likewise, assume

$$\begin{cases} p(\vec{x}_n|C_1) = \mathcal{N}(\vec{x}_n|\vec{\mu}_1, \Sigma) \\ p(\vec{x}_n|C_2) = \mathcal{N}(\vec{x}_n|\vec{\mu}_2, \Sigma) \end{cases}$$

shared

Estimate $\vec{\mu}_1, \vec{\mu}_2, \Sigma$ by ML

Then

$$\mathcal{J}' = \prod_{n=1}^{N} [\mathcal{N}(\vec{x}_n|\vec{\mu}_1, \Sigma)]^{t_n} [\mathcal{N}(\vec{x}_n|\vec{\mu}_2, \Sigma)]^{1-t_n},$$

$$\begin{aligned}\log \mathcal{J}' = &-\frac{1}{2}\sum_{n=1}^{N} t_n(\vec{x}_n-\vec{\mu}_1)^T\Sigma^{-1}(\vec{x}_n-\vec{\mu}_1) - \\ &-\frac{1}{2}\sum_{n=1}^{N}(1-t_n)(\vec{x}_n-\vec{\mu}_2)^T\Sigma^{-1}(\vec{x}_n-\vec{\mu}_2) - \\ &-\frac{1}{2}\sum_{n=1}^{N} t_n \log|\Sigma| - \frac{1}{2}\sum_{n=1}^{N}(1-t_n)\log|\Sigma|\end{aligned}$$

Omitted $-\frac{D}{2}\sum_{n=1}^{N} t_n \log(2\pi) - \frac{D}{2}\sum_{n=1}^{N}(1-t_n)\log(2\pi) = \underbrace{-\frac{DN}{2}\log(2\pi)}_{\text{const}},$

$$\frac{\partial}{\partial\vec{\mu}_1}\log\mathcal{J}' = 0 \Rightarrow \sum_n t_n\Sigma^{-1}(\vec{x}_n-\vec{\mu}_1) = 0,$$

$$\Sigma^{-1}\Big(\underbrace{\sum_n t_n(\vec{x}_n-\vec{\mu}_1)}_{=0}\Big) = 0$$

$$\vec{\mu}_1 = \frac{\sum_n t_n\vec{x}_n}{\underbrace{\sum_n t_n}_{N_1}} = \frac{1}{N_1}\sum_n t_n\vec{x}_n$$

Likewise, $\vec{\mu}_2 = \frac{1}{N_2}\sum_n(1-t_n)\vec{x}_n.$

Finally,

$$\log \mathcal{L}' = -\frac{N}{2}\log|\Sigma| - \frac{1}{2}\sum_{n\in C_1}(\vec{x}_n-\vec{\mu}_1)^T\Sigma^{-1}(\vec{x}_n-\vec{\mu}_1) - $$

$$-\frac{1}{2}\sum_{n\in C_2}(\vec{x}_n-\vec{\mu}_2)^T\Sigma^{-1}(\vec{x}_n-\vec{\mu}_2) + \mathrm{const}(\Sigma) \;\; \textcircled{=}$$

$$\sum_{n\in C_1}\sum_{i,j}(x_{n,i}-\mu_{1,i})\Sigma^{-1}_{ij}(x_{n,j}-\mu_{1,j}) =$$

$$=\sum_{ij}\Sigma^{-1}_{ij}\Big[\underbrace{\sum_{n\in C_1}(x_{n,j}-\mu_{1,j})(x_{n,i}-\mu_{1,i})}_{\overset{\shortparallel}{N_1 S_{1,ji}}}\Big] =$$

$$= \mathrm{Tr}(\Sigma^{-1}S_1)\cdot N_1.$$

$$\textcircled{=} -\frac{N}{2}\log|\Sigma| - \frac{N}{2}\mathrm{Tr}(\Sigma^{-1}[\underbrace{\frac{N_1}{N}S_1+\frac{N_2}{N}S_2}_{S}]) =$$

$$= -\frac{N}{2}\log|\Sigma| - \frac{N}{2}\mathrm{Tr}(\Sigma^{-1}S).$$

$$\frac{\partial}{\partial\Sigma_{ij}}\log\mathcal{L}' = -\frac{N}{2}\Sigma^{-1}_{ji} + \frac{N}{2}(\Sigma^{-1}S\Sigma^{-1})_{ji} = 0, \text{ so that}$$

$$\Sigma^{-1} = \Sigma^{-1}S\Sigma^{-1} \Rightarrow \boxed{S=\Sigma}$$

$$\underset{(C.28)}{\nearrow}\frac{\partial}{\partial\Sigma}\log|\Sigma| = \underbrace{\Sigma^{-1}}_{(\Sigma^{-1})^T} \leftarrow \frac{\partial}{\partial\Sigma_{ij}}\log|\Sigma| = (\Sigma^{-1})^T_{ij} = \Sigma^{-1}_{ji} =$$

$$= \Sigma^{-1}_{ij} \text{ since } \Sigma^{-1} \text{ is symm.}$$

$$\frac{\partial}{\partial\Sigma_{ij}}\mathrm{Tr}(\Sigma^{-1}S) = \mathrm{Tr}\Big(\frac{\partial}{\partial\Sigma_{ij}}(\Sigma^{-1})S\Big) =$$

$$= -\mathrm{Tr}\Big(\Sigma^{-1}\overset{\mathbb{I}_{ij}}{\frac{\partial\Sigma}{\partial\Sigma_{ij}}}\Sigma^{-1}S\Big) = -\mathrm{Tr}(\mathbb{I}_{ij}\Sigma^{-1}S\Sigma^{-1}) =$$

$\nearrow$ 1 at (i,j), 0 everywhere else [ignores $\Sigma_{ij}=\Sigma_{ji}$]
$\nearrow$ cf. 2.3.4

$$= -(\Sigma^{-1}S\Sigma^{-1})_{ji}.$$

Finally, $p(C_1|\vec{x}) = \sigma(\vec{w}^T\vec{x} + w_0)$, with

$\vec{w}$ & w_0 f's of $\vec{\mu}_1, \vec{\mu}_2, \Sigma$, which are all estimated <u>by ML</u>. $p(C_1|D)$

$$p(C_2|\vec{x}) = 1 - p(C_1|\vec{x}).$$

Alternatively, we do not have to use the Bayes formula and can find $p(C_k|\vec{x})$ more directly. This is often easier and may lead to better fits.

We will use a fixed non-linear transform into feature space: $\vec{\varphi}(\vec{x})$.

<u>Logistic regression</u>

Consider $\boxed{K=2}$ again:

$$\begin{cases} p(C_1|\vec{\varphi}) = y(\vec{\varphi}) = \sigma(\vec{w}^T\vec{\varphi}) \quad \uparrow \varphi_0(\vec{x}) = 1 \\ p(C_2|\vec{\varphi}) = 1 - p(C_1|\vec{\varphi}) \end{cases}$$

If $\vec{\varphi}$ has dimensions M, there are M fitting prms. Recall that in the previous approach, we need $2M$ prms to fit $\vec{\mu}_1$ & $\vec{\mu}_2$, $\frac{M(M+1)}{2}$ prms to fit $\overset{\text{shared cov. matrix}}{\Sigma}$, and 1 prm to fit $P(C_1)/P(C_2)$. So it's $O(M)$ vs. $O(M^2)$.

Consider $\{\vec{\varphi}_n, t_n\}$ $n = 1, \dots, N$

$t_n \in \{0, 1\}$ $\vec{\varphi}_n = \vec{\varphi}(\vec{x}_n) \Rightarrow \underbrace{\varphi_j(\vec{x}_n)}_{\equiv \varphi_{n,j}} = \varphi_{nj}$

Then $p(\vec{t}\,|\,\vec{w}) = \prod_{n=1}^{N} y_n^{t_n}(1-y_n)^{1-t_n}$

$\overset{\nearrow}{\mathcal{L}}$

$$\begin{cases} y_n = p(C_1|\vec{\varphi}_n), = \sigma(\underbrace{\vec{w}^T\vec{\varphi}_n}_{a_n}) \\ 1 - y_n = p(C_2|\vec{\varphi}_n). \end{cases}$$

The error f'n is given by

$$E(\vec{w}) = -\log p(\vec{t}\,|\,\vec{w}) = -\sum_{n=1}^{N}[t_n \log y_n + (1-t_n)\log(1-y_n)]$$

$$\frac{\partial E(\vec{w})}{\partial w_i} = -\sum_{n=1}^{N}\left[\frac{t_n}{y_n}\frac{\partial y_n}{\partial w_i} - (1-t_n)\frac{1}{1-y_n}\frac{\partial y_n}{\partial w_i}\right] =$$

$$= \sum_{n=1}^{N} \underbrace{\frac{(1-t_n)y_n - t_n(1-y_n)}{y_n(1-y_n)}}_{\frac{y_n - t_n}{y_n(1-y_n)}} \underbrace{\frac{\partial y_n}{\partial w_i}}_{y_n(1-y_n)\underbrace{\frac{\partial a_n}{\partial w_i}}_{\varphi_{n,i}}} \textcircled{=}$$

$$\textcircled{=} \underline{\underline{\sum_{n=1}^{N}(y_n - t_n)\varphi_{n,i}}}.$$

Difference between t_n & y_n multiplied by the basis f'n values.

Note that $E(\vec{w})$ is a non-linear f'n of $\vec{w}$ so there's no closed-form solution (unlike regression). However, one can argue that $E(\vec{w})$ is con~~[illegible]~~ vex everywhere:

↑ as shown below

has a unique minimum

We can find it using the Newton-Raphson (NR) algorithm:

$$\vec{\nabla} \equiv \frac{\partial}{\partial \vec{w}}$$

$$\vec{w}^{\,new} = \vec{w}^{\,old} - H^{-1} \vec{\nabla} E(\vec{w})$$

H is the Hessian matrix: $H_{ij} = \dfrac{\partial^2 E(\vec{w})}{\partial w_i \partial w_j}$

Let's first apply it to regression:

$$\tilde{E}(\vec{w}) = \frac{1}{2} \sum_{n=1}^{N} (t_n - \vec{w}^T \cdot \vec{\varphi}_n)^2$$ quadratic error f'n

Then $\dfrac{\partial \tilde{E}}{\partial w_i} = \sum_{n=1}^{N} (\vec{w}^T \cdot \vec{\varphi}_n - t_n)\, \varphi_{n,i} \Rightarrow$

$$\Rightarrow \frac{\partial \tilde{E}}{\partial \vec{w}} = (\varphi^T \varphi)\vec{w} - \varphi^T \vec{t}$$

$\sum_{n=1}^{N} t_n \varphi_{n,i} = \sum_n \varphi_{ni} t_n = \sum_n (\varphi^T)_{in} t_n = (\varphi^T \vec{t})_i$

$\sum_{n,j} w_j \varphi_{n,j} \varphi_{n,i} = \sum_n \varphi_{ni}\varphi_{nj} w_j = \sum_n (\varphi^T)_{in} \varphi_{nj} w_j = ((\varphi^T\varphi)\vec{w})_i$

Further,

$$\frac{\partial^2 \tilde{E}}{\partial w_i \partial w_j} = \sum_{n=1}^{N} \underbrace{\varphi_{n,j}\, \varphi_{n,i}}_{\varphi_{nj}\varphi_{ni} = (\varphi^T)_{in}\varphi_{nj}} \Rightarrow H = \varphi^T \varphi$$

Then $\vec{w}^{\,new} = \vec{w}^{\,old} - (\varphi^T\varphi)^{-1}\left[(\varphi^T\varphi)\vec{w}^{\,old} - \varphi^T \vec{t}\right] =$

$$= (\varphi^T\varphi)^{-1} \varphi^T \vec{t}$$ exact sol'n in one step

NR gives the exact solution b/c $\tilde{E}(\vec{w})$ is quadratic.

Now let's consider $E(\vec{w})$:

$$\vec{\nabla} E = \Phi^T(\vec{y}-\vec{t}) \quad \leftarrow \text{non-linear function of } \vec{w}$$

$$\vec{y} = \{y_1, \dots, y_N\} \qquad \vec{t} = \{t_1, \dots, t_N\}$$

$$\vec{H} = \vec{\nabla}\vec{\nabla} E \Rightarrow \frac{\partial^2 E}{\partial w_i \partial w_j} = \sum_{n=1}^{N} \frac{\partial y_n}{\partial w_j} \varphi_{n,i} =$$

$$= \sum_n y_n(1-y_n) \overset{\Phi_{nj}}{\widetilde{\varphi_{n,j}}} \overset{\Phi_{ni} = (\Phi^T)_{in}}{\widetilde{\varphi_{n,i}}} .$$

Then

$$H = \Phi^T R \Phi$$

$$\underset{\text{diagonal matrix}}{R} = \left.\begin{pmatrix} \overbrace{y_1(1-y_1) \quad\quad\quad\quad\quad\quad}^{N} \\ \ddots \\ \quad\quad\quad\quad\quad y_N(1-y_N) \end{pmatrix}\right\} N$$

Note that $H = H(\vec{w})$ through y_n.

since $E(\vec{w})$ is non-quadratic

Since $y_n(1-y_n) > 0$, $\forall n \Rightarrow H$ is positive definite.

But then $E(\vec{w})$ is convex everywhere & has a unique minimum.

Finally, NR gives

$$\vec{w}^{\text{new}} = \vec{w}^{\text{old}} - (\Phi^T R \Phi)^{-1} \Phi^T(\vec{y}-\vec{t}) =$$

$$= (\Phi^T R \Phi)^{-1} \left[(\Phi^T R \Phi) \vec{w}^{\text{old}} - \Phi^T(\vec{y}-\vec{t}) \right] =$$

$$= (\Phi^T R \Phi)^{-1} \Phi^T R \underbrace{\vec{z}}_{\Phi \vec{w}^{\text{old}} - R^{-1}(\vec{y}-\vec{t})} \leftarrow \text{N-dim. vector} \qquad (*)$$

Apply (*) iteratively to get $\vec{w}$.

Note that, as with regression,

$$E[t|\vec{x}] = y(\vec{x}) = \sigma(\underset{\uparrow \text{fitted weights}}{\vec{w}}^{T} \cdot \vec{\varphi}(\vec{x}))$$

$$Var[t|\vec{x}] = E[\underset{\overset{\|}{t \text{ since } t\in\{0,1\}}}{t^2}|\vec{x}] - E^2[t|\vec{x}] = y - y^2 = y(1-y).$$

So, elements of the R matrix are variances of t_n.

~~Moreover, note that for $a_n = \vec{w}^T \vec{\varphi}_n$~~

~~$a_n(\vec{w})$~~

Alternatively, consider

$$\vec{\nabla} E(\vec{w}) = \underbrace{\varphi^T}_{\substack{\#\text{prms} \to M \times N \leftarrow \#\text{datapoints} \\ \uparrow \quad \uparrow \\ \text{rows} \quad \text{columns}}} (\underbrace{\vec{y} - \vec{t}}_{N \text{ vector}}) = 0 \quad \text{directly.}$$

note that here $\varphi^T \vec{y} = \varphi^T \vec{t}$ leads to
$\vec{y} = \underbrace{(\varphi^T)^+}_{\substack{\text{Moore-Penrose pseudoinverse} \\ = (\varphi\varphi^T)^{-1}\varphi}} \varphi^T \vec{t}$, same as below

Then $$(\underbrace{\varphi\varphi^T}_{[N\times M * M\times N] = N\times N})\vec{y} = (\varphi\varphi^T)\vec{t}, \text{ or}$$

$$\vec{y} = \overbrace{(\varphi\varphi^T)^{-1}(\varphi\varphi^T)}^{= \mathbb{I}}\vec{t} \Rightarrow \underline{\underline{\vec{y} = \vec{t}}}, \text{ as expected}$$

$$\vec{y} = \{\sigma(a_1) \dots \sigma(a_N)\}$$

In components,

$$y_n = \sigma(a_n) = t_n$$

But then

$$a_n = \log\left(\frac{t_n}{1-t_n}\right) \equiv C_n$$

C_n: component of an N vector with $\{+\infty, -\infty\}$ components (!)

$$a_n = \sum_{j=1}^{M} w_j \underbrace{\varphi_{n,j}}_{= \varphi_j(\vec{x}_n) = \varphi_{nj}} = \sum_j \varphi_{nj} w_j$$

Finally, in matrix form

$$\underbrace{\underbrace{\varphi}_{N\times M} \underbrace{\vec{w}}_{M\text{vec}}}_{} = \underbrace{\vec{C}}_{N\text{ vec}} \Rightarrow \underbrace{\vec{w}}_{M\text{vec}} = \underbrace{\varphi^+}_{M\times N} \underbrace{\vec{C}}_{N\text{ vec}} = (\varphi^T\varphi)^{-1}\varphi^T\vec{C}$$

$$(\varphi^T\varphi)\vec{w} = \varphi^T\vec{C} \quad N \text{ vector}$$

Clearly at least some weights will be $\pm\infty$, but the solution is unique