ML solution

Lecture 8

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

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

Before we assumed that $p(\vec{x}|C_i)$ is a 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 $K=2$ for simplicity.

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

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

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

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

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

$$\pi N = \sum_{n=1}^{N} t_n \Rightarrow \pi = \frac{1}{N} \sum_{n=1}^{N} t_n$$

" $p(C_1|\mathcal{D})_{ML}$

Likewise, assume (shared)

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

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

Then

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

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

$$- \frac{1}{2} \sum_{n=1}^{N} (1-t_n) (\vec{x}_n - \vec{\mu}_2)^T \Sigma^{-1} (\vec{x}_n - \vec{\mu}_2) -$$

$$- \frac{1}{2} \sum_{n=1}^{N} t_n \log|\Sigma| - \frac{1}{2} \sum_{n=1}^{N} (1-t_n) \log|\Sigma| .$$

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

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

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

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

Finally,

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

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

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

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

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

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

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

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

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

$$\frac{\partial}{\partial\Sigma}\log|\Sigma| = \underbrace{\Sigma^{-1}}_{\text{"}(\Sigma^{-1})^T} \quad \text{(C.28)}$$

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

$$\frac{\partial}{\partial\Sigma_{ij}} = \mathbb{I}_{ij}$$

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

$\mathbb{I}_{ij}$: 1 at (i,j), 0 everywhere else [ignores $\Sigma_{ij}=\Sigma_{ji}$]

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

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 by ML.

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

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: $\vec{\varphi}(\vec{x})$.

Logistic regression

Consider $K=2$ again:

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

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 Σ (shared cov. matrix), and 1 prm to fit $P(C_1)/P(C_2)$. So it's $\Theta(M)$ vs. $\Theta(M^2)$.

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

$t_n \in \{0, 1\}$ $\vec{\varphi}_n = \vec{\varphi}(\vec{x}_n)$

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

$\nearrow$ likelihood

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

The error f'n is given by

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

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

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

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

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 convex everywhere:

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: $H_{ij} = \frac{\partial E(\vec{w})}{\partial w_i \partial w_j}$

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

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

Further,

$$\frac{\partial \tilde{E}}{\partial w_i \partial w_j} = \sum_{n=1}^{N} \varphi_{n,j} \varphi_{n,i} \Rightarrow H = \Phi^T \Phi$$

Then $\vec{w}^{new} = \vec{w}^{old} - (\Phi^T \Phi)^{-1} [(\Phi^T \Phi) \vec{w}^{old} - \Phi^T \vec{t}] =$

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

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

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

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

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

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

$$= \sum_n y_n (1 - y_n) \varphi_{n,j} \varphi_{n,i} .$$

Then

$$H = \varphi^T R \varphi$$

$$R = \begin{pmatrix} y_1(1-y_1) & & \\ & \ddots & \\ & & y_N(1-y_N) \end{pmatrix} \Big\} N \quad \text{(diagonal matrix)}$$

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

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

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

$$= (\varphi^T R \varphi)^{-1} \varphi^T R \vec{z} \quad (*)$$

($\vec{z}$: N-dim. vector) $= \varphi \vec{w}^{old} - R^{-1} (\vec{y} - \vec{t})$

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

Note that, as with regression,

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

($\vec{w}$: fitted weights)

$$\text{Var}[t|\vec{x}] = E[t^2|\vec{x}] - E^2[t|\vec{x}] = y - y^2 = y(1-y).$$

($E[t^2|\vec{x}] = t$ since $t \in \{0,1\}$)

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

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

(φ^T: M = # prims, N = # datapoints; $(\vec{y}-\vec{t})$: N vector)

note that here $\varphi^T \vec{y} = \varphi^T \vec{t}$ leads to

$$\vec{y} = \underbrace{(\varphi^T)^+}_{\text{Moore-Penrose pseudoinverse}} \varphi^T \vec{t}$$

$= (\varphi\varphi^T)^{-1}\varphi$, same as below

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

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

as expected

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

In components,

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

But then $a_n = \log\left(\frac{t_n}{1-t_n}\right) \equiv C_n$ (component of an N vector with $\{+\infty, -\infty\}$ components (!))

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

Finally, in matrix form

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

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

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