5. Modelos lineares para classificação

No Capítulo 4, você aprendeu a utilizar modelos lineares para resolver problemas simples de regressão. Você também aprendeu como estender regressão linear para ajustar curvas complexas e construir redes neurais simples. Nesse Capítulo, desenvolveremos modelos lineares para o caso em que a variável de resposta y é uma *classe*, ao invés de um número real.

5.1 Regressão logística

Uma das maneiras mais naturais de criar um modelo de regressão consiste em escolher um modelo observacional para a variável de resposta y cujos parâmetros dependam diretamente do seu respectivo vetor de entradas $\boldsymbol{x}$. Por exemplo, na capítulo anterior, vimos que minimizar o MSE é equivalente a admitir uma verossimilhança da forma $y|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\theta}^\intercal \boldsymbol{x}, \sigma^2)$, na qual o parâmetro de média é uma função linear de $\boldsymbol{x}$. Note também que esse escolha implica que y pode tomar valores arbitrários em $\mathbb{R}$, já que esse é o suporte da distribuição normal (i.e., região com densidade maior que zero).

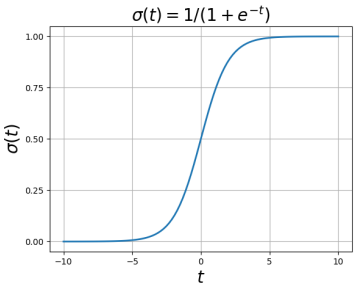
Se y é uma variável binária (0/1), a escolha mais comum é utilizar uma distribuição Bernoulli com parâmetro $r = g(\boldsymbol{x})$. Em outras palavras, falamos que y assume valor 1 com probabilidade r e 0 com probabilidade $1 - r$. Adotando esse modelo observacional para $y|\boldsymbol{x}$, resta-nos definir a função g para completar nosso modelo de regressão. Para tal, iremos calcular uma função linear de $\boldsymbol{x}$, como anteriormente, mas aplicaremos uma função que mapeie o valor resultante (comumente conhecido como logit) para $[0, 1]$, gerando valores válidos para a probabilidade r. Mais especificamente, usamos a função sigmoide $\sigma(t) = (1 + e^{-t})^{-1}$ para definir a probabilidade de $y|\boldsymbol{x}$ como:

$$p(y|\boldsymbol{x}) = \text{Ber}\left(y|\sigma\left(\boldsymbol{\theta}^\intercal \boldsymbol{x}\right)\right) = \sigma\left(\boldsymbol{\theta}^\intercal \boldsymbol{x}\right)^y \left(1 - \sigma\left(\boldsymbol{\theta}^\intercal \boldsymbol{x}\right)\right)^{1-y}. \tag{5.1}$$

Propriedades da função sigmoide. A Figura abaixo ilustra a função sigmoide. Note que ela é estritamente crescente, i.e., $t < t'$ implica $\sigma(t) < \sigma(t')$. Notavelmente, $\sigma(t)$ se aproxima de 0 quando t diminui e se aproxima de 1 quando t aumenta.

A função sigmoide também possui propriedades analíticas especialmente úteis computacionalmente, por exemplo:

$$\sigma(-t) = 1 - \sigma(t), \qquad \frac{\mathrm{d}\,\sigma(t)}{\mathrm{d}\,t} = \sigma(t)\sigma(-t),$$
$$\sigma(t) = \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{t}{2}\right), \qquad \int \sigma(t)\mathrm{d}\,t = \log\sigma(-t) + C.$$



Dado um conjunto de treinamento $\mathcal{D}$ com N exemplos de treinamento $(\boldsymbol{x}_n, y_n)$, podemos então definir a função de verossimilhança $\mathcal{L}$ como

$$\mathcal{L}(\boldsymbol{\theta}) = \prod_{i=1}^{N} \sigma\left(\boldsymbol{\theta}^\intercal \boldsymbol{x}_i\right)^{y_i} \left(1 - \sigma\left(\boldsymbol{\theta}^\intercal \boldsymbol{x}_i\right)\right)^{1-y_i}, \tag{5.2}$$

e otimizá-la para encontrar a estimativa de máxima verossimilhança para $\boldsymbol{\theta}$:

$$\begin{aligned}
\hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}\in\mathbb{R}^{D+1}} \mathcal{L}(\boldsymbol{\theta}) \equiv \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^{D+1}} -\log \mathcal{L}(\boldsymbol{\theta}) \\
&\equiv \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^{D+1}} -\sum_{i=1}^{N} y_i \log \sigma\left(\boldsymbol{\theta}^\intercal \boldsymbol{x}_i\right) + (1 - y_i) \log\left(1 - \sigma\left(\boldsymbol{\theta}^\intercal \boldsymbol{x}_i\right)\right) \\
&\equiv \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^{D+1}} -\sum_{\substack{i=1\ldots N \\ y_i=1}} \log \sigma\left(\boldsymbol{\theta}^\intercal \boldsymbol{x}_i\right) - \sum_{\substack{i=1\ldots N \\ y_i=0}} \log \sigma\left(-\boldsymbol{\theta}^\intercal \boldsymbol{x}_i\right)
\end{aligned}$$

Similar ao MSE, que minimizamos no capítulo passado, a função objetivo $-\log \mathcal{L}$ é uma convexa. No entanto, não possuímos uma solução analítica para $\hat{\boldsymbol{\theta}}$ e, portanto, precisamos utilizar algum método de otimização númerica, como o SGD que já conhecemos. Para fins de implementação, podemos definir a variável $y_i' = 2y_i - 1$. Com isso, conseguimos escrever o gradiente $\nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta})$ da log verossimilhança para o i-ésimo exemplo de treinamento ℓ_i como:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \log \sigma(y_i' \boldsymbol{\theta}^\intercal \boldsymbol{x}_i) \\
&= \frac{\partial \log \sigma(y_i' \boldsymbol{\theta}^\intercal \boldsymbol{x}_i)}{\partial \sigma(y_i' \boldsymbol{\theta}^\intercal \boldsymbol{x}_i)} \frac{\partial \sigma(y_i' \boldsymbol{\theta}^\intercal \boldsymbol{x}_i)}{\partial y_i' \boldsymbol{\theta}^\intercal \boldsymbol{x}_i} \frac{\partial y_i' \boldsymbol{\theta}^\intercal \boldsymbol{x}_i}{\partial \boldsymbol{\theta}} \\
&= \sigma(-y_i' \boldsymbol{\theta}^\intercal \boldsymbol{x}_i) y_i' \boldsymbol{x}_i
\end{aligned}$$

Aplicando o algoritmo gradiente descendente à função custo da regressão logística, obtemos o seguinte algoritmo:

Algoritmo 1 Regressão Logística

1: $\boldsymbol{\theta}^{(0)} \leftarrow \mathbf{0}$ ▷ Inicializa os parâmetros/pesos.
2: $\boldsymbol{y}' \leftarrow 2\boldsymbol{y} - \mathbf{1}$ ▷ Transforma os rótulos para -1 e 1.
3: **for** $t = 0, 1, 2 \ldots$ **do**
4: $\quad \boldsymbol{g} \leftarrow \sum_i \sigma(-y_i' \boldsymbol{x}_i^\intercal \boldsymbol{\theta}^{(t)}) y_i' \boldsymbol{x}_i$ ▷ Computa o gradiente.
5: $\quad \boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \eta \boldsymbol{g}$ ▷ Realiza a atualização dos parâmetros.
6: $\quad$ Verifica a condição de parada. Por exemplo, se $\|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\| < 0.000001$.
7: **end for**
8: **return** $\boldsymbol{\theta}$ ▷ Retorna os parâmetros/pesos finais.

Interpretação geométrica. Uma vez que obtivemos $\hat{\boldsymbol{\theta}}$, podemos estimar a probabilide de uma nova amostra $\boldsymbol{x}^\star$ pertencer à classe 1 como $\sigma(\hat{\boldsymbol{\theta}}^\intercal \boldsymbol{x}^\star)$ e a de pertencer à classe 0 como $1 - \sigma(\hat{\boldsymbol{\theta}}^\intercal \boldsymbol{x}^\star)$. Com isso em mente, se precisamos prever a classe de $\boldsymbol{x}^\star$, é razoável escolher aquela que achamos mais provável. Lembre que $\sigma(0) = 0.5$. Portanto, o plano $\hat{\boldsymbol{\theta}}^\intercal \boldsymbol{x} = 0$ caracteriza os pontos $\boldsymbol{x} \in \mathcal{X}$ que cremos ter probabilidade idêntica de pertencer a ambas as classes. Isso implica que todos os vetores de entrada cujo ângulo γ com o vetor

normal $\hat{\boldsymbol{\theta}}$ é menor que noventa graus são classificados como positivos (classe 1) — lembre que $\hat{\boldsymbol{\theta}}^\top \boldsymbol{x} = \|\hat{\boldsymbol{\theta}}\|_2 \|\boldsymbol{x}\|_2 \cos\gamma$. Os demais pontos são classificados como negativos (classe 0).

Convexidade do problema de aprendizado. Como a soma de funções convexas é também convexa, basta verificar que as $\ell_1, \ldots, \ell_N$ são convexas para provarmos que $-\log \mathcal{L}$ também o é. Para esse fim, podemos usar o fato de que a função composta $h = g \circ f$ é convexa se f é côncava e g é convexa não-crescente. Note que $y_i' \boldsymbol{\theta}^\top \boldsymbol{x}_i$ é tanto côncava como convexa, como é o caso de funções lineares. Em contrapartida, $-\log \sigma(t)$ é convexa não-crescente já que ela descresce com t e sua derivada $-\sigma(-t)$ é estritamente crescente.

Entropia cruzada binária. Na comunidade de ML, é comum se referir ao logaritmo negativo da verossimilhança Bernoulli como entropia cruzada binária (*binary cross entropy*, BCE). De forma geral, a entropia cruzada entre duas funções de massa/densidade p e q sobre a mesma variável aleatória z e com suportes idênticos é definida como:

$$H(p, q) = \mathbb{E}_{z \sim p}\left[\log \frac{1}{q(z)}\right],$$

e dá-se o nome BCE para o caso especial em que p e q são distribuições Bernoulli.

Em teoria da informação, é comum interpretar $\log 1/q(z)$ como uma medida de *surpresa*, i.e., do quanto observar um valor específico z contrasta com seu conhecimento prévio, representado por q. Nesse contexto, $H(p, q)$ é o valor dessa medida se os valores de z são amostrados de p (ao invés de q). Vale ressaltar que, para um p fixo, $q = p$ minimiza $H(p, q)$. Nesse caso, a quantia $H(p) \coloneqq H(p, p)$ é chamada de entropia. Por sua vez, a entropia também pode ser vista como uma medida de concentração de p, atingindo seu valor máximo quando p é uma distribuição uniforme.

Para concluir que a BCE generaliza $-\log \mathrm{Ber}(y|r)$, basta definir $p(z) = \mathrm{Ber}(z|y)$ e tomar $q(z) = \mathrm{Ber}(z|r)$. Com essas escolhas, obtemos:

$$\begin{aligned} H(p, q) &= -y \log q(1) - (1 - y) \log q(0) \\ &= -\left(y \log r + (1 - y) \log (1 - r)\right) \end{aligned}$$

o que implica que:

$$e^{-H(p,q)} = r^y (1 - r)^{(1-y)} = \mathrm{Ber}(y|r)$$

5.2 Regressão logística Bayesiana

Na seção anterior, discutimos como obter uma estimativa pontual $\hat{\boldsymbol{\theta}}$ para $\boldsymbol{\theta}$ via máxima verossimilhança (MLE). Um problema com estimativas pontuais, no entanto, é que elas não refletem nossa incerteza sobre o valor estimado. Intuitivamente, por exemplo, esperamos que estimativas feitas com grandes quantidades de dados sejam mais confiáveis que as feitas com poucos dados. Para observar que MLE não captura esse aspecto, basta notar que repetir o conjunto de dados qualquer número arbitrário de vezes não causa impacto algum em $\hat{\boldsymbol{\theta}}$.

A estatística Bayesiana propõe uma saída intuitiva para esse empasse. A ideia é modelar os parâmetros $\boldsymbol{\theta}$ do modelo como uma variável aleatória, usando uma distribuição *a priori* $p(\boldsymbol{\theta})$ e

aplicar a regra de Bayes para computar a distribuição de $\boldsymbol{\theta}$ condicionada nas observações $\mathcal{D}$, que chamamos de *posteriori*:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}} p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}$$

Informalmente, a posteriori representa a incerteza que temos sobre o valor da variável $\boldsymbol{\theta}$. Além disso, vale ressaltar que a priori nos permite encapsular conhecimento prévio, potencialmente subjetivo, sobre $\boldsymbol{\theta}$ (i.e., antes de ver os dados) e sua escolha é uma questão de modelagem estatística. Quando não possuímos informação significante sobre os dados, é comum escolher uma distribuição de alta entropia como priori. No caso em que $\boldsymbol{\theta}$ assume valores reais, e.g., poderíamos usar uma priori Gaussiana com alta variância.

No caso da regressão logística, é comum adotar uma priori Gaussiana sobre $\boldsymbol{\theta}$ e a verossimilhança que usamos na seção anterior, resultando no modelo:

$$\begin{aligned}\boldsymbol{\theta} &\sim \mathcal{N}(\mu, \Sigma)\\ y_n \,|\, \boldsymbol{x}_n &\sim \text{Ber}(\sigma(\boldsymbol{\theta})^\intercal \boldsymbol{x}_n) \forall n = 1 \dots N,\end{aligned}$$

cuja posteriori é dada por

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{\prod_{i=1}^{N} \text{Ber}(y_i|\sigma(\boldsymbol{\theta}^\intercal \boldsymbol{x}_i))\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})}{\int_{\boldsymbol{x}\in\mathbb{R}^{D+1}} \prod_{i=1}^{N} \text{Ber}(y_i|\sigma(\boldsymbol{\theta}^\intercal \boldsymbol{x}_i))\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})} \tag{5.3}$$

Notavelmente, computar a posteriori acima depende da resolução de uma integral que não possui forma fechada. Consequentemente, também não há uma forma analítica para $p(\boldsymbol{\theta}|\mathcal{D})$. Essa dificuldade técnica não é uma raridade em modelos Bayesianos. Apesar de algumas escolhas pareadas de verossimilhança e priori resultarem em posteriores com forma analítica, esse não é o caso geral. Para driblar esse problema, usaremos métodos numéricos para aproximar a posteriori com distribuição mais simples, de forma conhecida.

Prevendo a label $y^\star$ para um novo input $\boldsymbol{x}^\star$. Suponha que conseguimos computar a posteriori $p(\boldsymbol{\theta}|\mathcal{D})$, como podemos obter uma distribuição para $p(y^\star|\boldsymbol{x}^\star)$? Quando estavamos usando uma estimativa pontual $\hat{\boldsymbol{\theta}}$ (MLE), obtívemos uma distribuição sobre $y^\star$ simplesmente encaixando $\hat{\boldsymbol{\theta}}$ na nossa verossimilhança, i.e., tomamos $p(y^\star|\boldsymbol{x}^\star) \approx \text{Ber}(\sigma(\boldsymbol{\theta}^\intercal \boldsymbol{x}^\star))$. No paradigma Bayesiano, levamos em consideração a incerteza sobre $\boldsymbol{\theta}$ (codificada em nossa posteriori), ponderando cada valoração de $\boldsymbol{\theta}$ pela sua densidade posteriori. O resultado, é o que chamamos de posteriori preditiva:

$$p(y^\star|\boldsymbol{x}^\star) = \int_{\boldsymbol{\theta}\in\mathbb{R}^{D+1}} p(y^\star|\boldsymbol{x}^\star, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})$$

Máxima verossimilhança e máximo *a posteriori*. Uma das maiores virtudes do paradigma Bayesiano é oferecer uma maneira de quantificar incerteza. No entanto, há situações nas quais computar a posteriori, mesmo que de maneira aproximada, pode se tornar computacionalmente indesejável. Nesses casos, é comum procurar o ponto que máximiza a posteriori e tomá-lo como estimativa pontual. Chama-se esse precedimento de máximo a

posteriori (MAP). Mais concretamente, a estimativa $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ pode ser escrita obtida como:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\text{MAP}} &= \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\mathcal{D}) \\ &= \arg\max_{\boldsymbol{\theta}} \log p(\mathcal{D}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \log \int_{\boldsymbol{\theta}} p(\mathcal{D}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \\ &\arg\max_{\boldsymbol{\theta}} \log p(\mathcal{D}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\end{aligned}$$

e, portanto, pode ser interpretada como uma versão *regularizada* do MLE, na qual $\log p(\boldsymbol{\theta})$ penaliza regiões pouco prováveis a priori.

Exemplo 5.1 — Priori conjugada. Para escolhas específicas de priori e verossimilhança, a distribuição posteriori possui forma analítica. Uma instância dessas ocorre quando $p(\theta)$ é uma distribuição Beta e a verossimilhança $p(\mathcal{D}|\theta)$ é Bernoulli. Mais concretamente, suponha que escolhemos uma priori $\text{Beta}(\alpha, \beta)$ para θ, dada por:

$$\text{Beta}(\theta|\alpha,\beta) = \frac{\theta^{\alpha-1}\theta^{\beta-1}}{B(\alpha,\beta)},$$

onde $B(\alpha,\beta) = \int_{\theta\epsilon[0,1]} \theta^{\alpha-1}\theta^{\beta-1}$ é uma constante normalizadora.

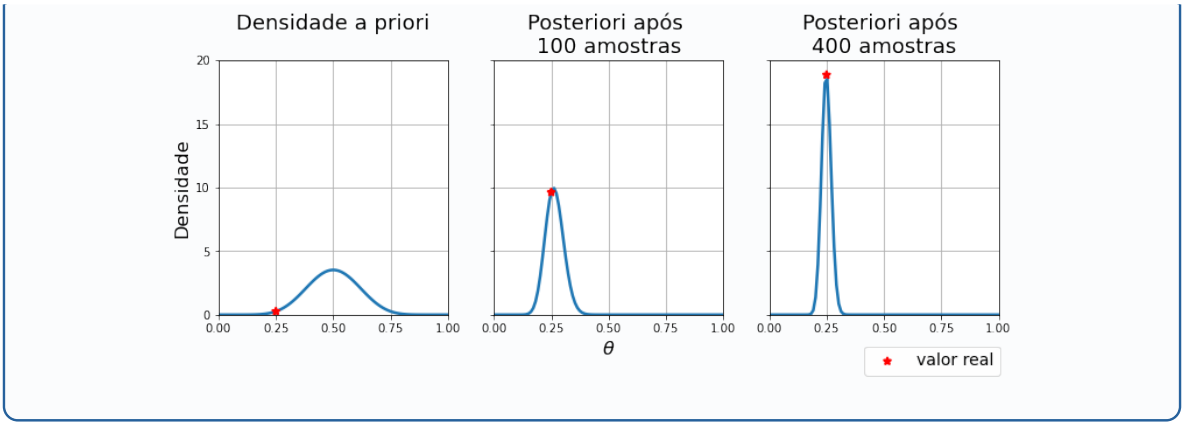
Podemos inferir, então, o seguinte sobre a posteriori do nosso modelo:

$$\begin{aligned}p(\theta|\mathcal{D}) &\propto \prod_{n=1}^{N} \text{Ber}(y_n|\theta)\text{Beta}(\theta|\alpha,\beta) \\ &\propto \prod_{n=1}^{N} \theta^{y_n}(1-\theta)^{1-y_n}\theta^{\alpha-1}\theta^{\beta-1} \\ &\propto \theta^{\sum_n y_n}(1-\theta)^{N-\sum_n y_n}\theta^{\alpha-1}\theta^{\beta-1} \\ &\propto \theta^{\sum_{n=1}^{N} y_n+\alpha-1}(1-\theta)^{N-\sum_{n=1}^{N} y_n+\beta-1} \\ &\propto \theta^{\alpha^\star-1}(1-\theta)^{\beta^\star-1}\end{aligned}$$

onde $\alpha^\star = \sum_{n=1}^{N} y_n + \alpha$ e $\beta^\star = N - \sum_{n=1}^{N} y_n + \beta$.

Note que $p(\theta|\mathcal{D})$ é proporcional à uma distribuição beta com parâmetros $\alpha^\star$ e $\beta^\star$. Portanto, concluímos que nossa posteriori é uma $\text{Beta}(\alpha^\star, \beta^\star)$. Chamamos casos como esse, no qual a posteriori pertence à mesma família da priori, de priori conjugada.

Para ilustrar o uso da regra de Bayes, a figura abaixo mostra atualizações da posteriori derivada acima para diferentes números de amostras N. Para tal, assumimos que a distribuição geradora dos dados é $\text{Ber}(0.25)$ e usamos uma priori $\text{Beta}(10, 10)$. Veja que, à medida que vemos mais amostras, a posteriori se afunila ao redor de 0.25.



5.2.1 Aproximação de Laplace

A aproximação de Laplace é, possivelmente, a mais simples técnica de inferência Bayesiana aproximada. A ideia é construir uma aproximação simples $q(\boldsymbol{\theta})$ para a posteriori $p(\boldsymbol{\theta} \mid \mathcal{D})$ usando uma expansão de Taylor de segunda ordem em $\log p(\boldsymbol{\theta} \mid \mathcal{D})$ ao redor da moda $\boldsymbol{m}$ da posteriori (i.e., o ponto de máxima densidade):

$$\log p(\boldsymbol{\theta} \mid \mathcal{D}) \approx \log p(\boldsymbol{m} \mid \mathcal{D}) + (\boldsymbol{\theta} - \boldsymbol{m})^\intercal \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} \mid \mathcal{D})\big|_{\boldsymbol{\theta}=\boldsymbol{m}} + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{m})^\intercal \nabla^2_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} \mid \mathcal{D})\big|_{\boldsymbol{\theta}=\boldsymbol{m}} (\boldsymbol{\theta} - \boldsymbol{m}),$$

onde ∇ denota o vetor gradiente e ∇^2 denota a matriz de segundas derivadas (Hessiana). Note que $\log p(\boldsymbol{m} \mid \mathcal{D})$ é uma constante com respeito a $\boldsymbol{\theta}$ e lembre que o gradiente de uma função em sua moda, caso ela exista, é zero. Então, concluímos que

$$\log p(\boldsymbol{\theta} \mid \mathcal{D}) \approx -\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{m})^\intercal \boldsymbol{H} (\boldsymbol{\theta} - \boldsymbol{m}) + \textit{constante},$$

onde $\boldsymbol{H} = \nabla^2_{\boldsymbol{\theta}} - \log p(\boldsymbol{\theta} \mid \mathcal{D})|_{\boldsymbol{\theta}=\boldsymbol{m}}$. Por *design*, construímos q tal que $\log q$ difira da expansão acima apenas por uma constante aditiva, então:

$$\log q(\boldsymbol{\theta}) = -\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{m})^\intercal \boldsymbol{H} (\boldsymbol{\theta} - \boldsymbol{m}) + \textit{constante} \Rightarrow q(\boldsymbol{\theta}) \propto e^{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{m})^\intercal \boldsymbol{H} (\boldsymbol{\theta}-\boldsymbol{m})},$$

e como $q(\boldsymbol{\theta})$ é proporcional á uma densidade normal multivariada com média $\boldsymbol{m}$ e matrix de covariância igual á inversa de $\boldsymbol{H}$, temos

$$q(\boldsymbol{\theta}) = \mathcal{N}\left(\boldsymbol{\theta} \mid \mu = \boldsymbol{m}, \boldsymbol{\Sigma} = \boldsymbol{H}^{-1}\right), \text{ onde } \boldsymbol{m} = \arg\max_{\boldsymbol{\theta} \in \Theta} p(\boldsymbol{\theta} \mid \mathcal{D}) \text{ e } \boldsymbol{H} = \nabla^2_{\boldsymbol{\theta}} - \log p(\boldsymbol{\theta} \mid \mathcal{D})\big|_{\boldsymbol{\theta}=\boldsymbol{m}}$$

Para aplicar o método de Laplace ao nosso modelo de regressão logística Bayesiano (Equação 5.3), podemos usar alguma variação de gradiente descendente para achar $\boldsymbol{m}$ e, assumindo $\boldsymbol{\mu} = \boldsymbol{0}$ e $\Sigma = cI$ para $c > 0$, as entradas $H_{i,j}$ da Hessiana $\boldsymbol{H}$ são dadas por:

$$H_{i,j} = \begin{cases} \sum_{n=1}^{N} \sigma(\boldsymbol{\theta}^\intercal \boldsymbol{x}_n)\sigma(-\boldsymbol{\theta}^\intercal \boldsymbol{x}_n) x_{n,i} x_{n,j} & \text{if } i \neq j, \\ \sum_{n=1}^{N} \sigma(\boldsymbol{\theta}^\intercal \boldsymbol{x}_n)\sigma(-\boldsymbol{\theta}^\intercal \boldsymbol{x}_n) x_{n,i} x_{n,j} + c^{-1} & \text{if } i = j. \end{cases}$$

5.2.2 Inferência variacional e o truque da reparametrização

Similar à aproximação de Laplace, a ideia por trás de inferência variacional é aproximar uma distribuição complexa p — uma posteriori, no nosso caso — com uma distribuição mais simples q. No entanto, ao invés de fazermos isso através de uma expansão de Taylor, escolhemos q dentro de uma família de distribuições Q de maneira a minimizar uma medida de *discrepância* para p. De forma geral, definimos a divergência de Kullback-Leibler (KL) entre p e q como:

$$D_{\mathrm{KL}}\left(q \| p\right) = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})}\right] = \int_{\Theta} q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} \, \mathrm{d}\boldsymbol{\theta}, \tag{5.4}$$

e encontramos uma aproximação $\hat{q}$ ótima minimizando a quantia acima sobre todo $q \in Q$, i.e.:

$$\hat{q} = \arg\min_{q \in Q} D_{\mathrm{KL}}\left(q \| p\right). \tag{5.5}$$

Note que a divergência KL é zero somente quando $q = p$. Apesar disso, de forma geral, a divergência KL não é simétrica — i.e., $D_{\mathrm{KL}}(p|q) \neq D_{\mathrm{KL}}(q\|p)$. Conseguimos também provar facilmente que a divergência KL é não negativa aplicando diretamente a desigualdade de Jensen.

No contexto de inferência Bayesiana, p costuma ser a posteriori $p(\boldsymbol{\theta}|\mathcal{D})$, que desejamos aproximar. Então, substituindo p na Equação 5.4 por $p(\boldsymbol{\theta} \,|\, \mathcal{D})$, obtemos:

$$D_{\mathrm{KL}}\left(q(\boldsymbol{\theta}) \| p(\boldsymbol{\theta} \,|\, \mathcal{D})\right) = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log q(\boldsymbol{\theta})\right] - \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log p(\boldsymbol{\theta}, \mathcal{D})\right] + \log p(\mathcal{D}), \tag{5.6}$$

onde $\log p(\boldsymbol{\theta}, \mathcal{D})$ pode ser escrito como $\log p(\boldsymbol{\theta}) + \log p(\mathcal{D}|\boldsymbol{\theta})$.

Como $\log p(\mathcal{D})$ é constante, definindo $-L(q) = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log q(\boldsymbol{\theta})\right] - \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log p(\boldsymbol{\theta}, \mathcal{D})\right]$, temos:

$$\hat{q} = \arg\min_{q \in Q} D_{\mathrm{KL}}\left(q \| p\right) = \arg\max_{q \in Q} L(q). \tag{5.7}$$

Em outras palavras, minimizar a divergência KL é equivalente a maximizar L. Vale ressaltar que a função L é popularmente conhecida como Evidence Lower BOund (ELBO), por ser um limitante inferior para o logaritmo da constante $p(\mathcal{D})$ no denominador da regra de Bayes — comumente chamada de evidência. Para perceber isso, basta combinar a não-negatividade da divergência KL com a Equação 5.6.

Com relação à escolha de Q, costuma-se adotar uma família de distribuições paramétricas. Deste modo, podemos maximizar L sobre um espaço de parâmetros Ω. Por exemplo, se Q é o conjunto das distribuições gaussianas univariadas, $\Omega = \mathbb{R} \times \mathbb{R}^+$ e $\boldsymbol{\omega} \in \Omega$ é um par média/variância. De maneira geral, os termos envolvidos no ELBO podem ser intratáveis e, até meados da década passada, desenvolver soluções customizadas para posteriore diferentes era considerado uma contribuição técnica em ML (i.e., virava paper ☺).

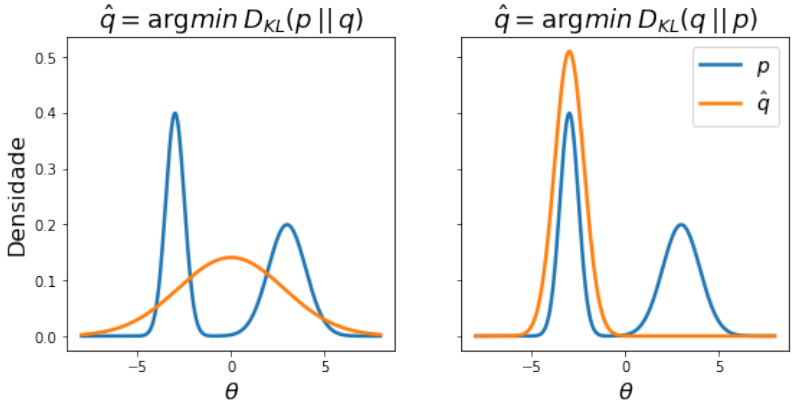
No entanto, existem agora metodologias genéricas, que viabilizam inferência variacional quase como uma tecnologia *off-the-shelf*. A mais famosa dentre essas, é o truque reparametrização. Essa técnica assume que é possível descrever a amostragem de $\boldsymbol{\theta} \sim q$ a partir de uma transformação g de uma variável aleatória auxiliar ϵ. Além, precisamos que q seja diferenciável com respeito aos parâmetros $\boldsymbol{\omega}$ de q. Por exemplo, se q é uma distribuição normal com parâmetros $\boldsymbol{\omega} = (\mu, \sigma^2)$, podemos obter uma amostra $\boldsymbol{\theta} \sim q$ definindo $g(\epsilon; \boldsymbol{\omega} = (\mu, \sigma^2)) = \epsilon\sigma + \mu$ e amostrando ϵ de numa gaussiana padrão — i.e., com média zero e variância um.

Com isso, podemos aproximar os termos do ELBO com M amostrar M variáveis auxiliares $\epsilon^{(1)}, \ldots, \epsilon^{(M)}$ e estimar o gradiente de $L(q)$ com respeito a $\boldsymbol{\omega}$ como:

$$\nabla_{\boldsymbol{\omega}} L(q) \approx \nabla_{\boldsymbol{\omega}} \frac{1}{M} \sum_{m=1}^{M} \log p(\mathcal{D}|\boldsymbol{\theta}^{(m)}) + \log p(\boldsymbol{\theta}^{(m)}) - \log q(\boldsymbol{\theta}^{(m)}),$$

onde $\boldsymbol{\theta}^{(m)} = g(\epsilon^{(m)}; \boldsymbol{\omega})$ — note que, na notação acima, q também depende diretamente de $\boldsymbol{\omega}$. Em posse dessa estimativa, podemos utilizar nosso algoritmo de gradiente preferido para otimizar o ELBO. Naturalmente, essa aproximação deve ser refeita, como novas amostras, a cada passo de gradiente. Para evitar incluir restrições explícitas para garantir que parâmetros restritos sejam válidos (e.g., variâncias devem ser não-negativas), nós também as modelamos como a transformação de uma variável real. No caso citado acima, podemos ter $\boldsymbol{\omega} = (\mu, \sigma^2 = u(z))$ com $u(z) = e^z$ ou $u(z) = \beta^{-1} \log(1 + e^{\beta z})$ para algum $\beta > 0$.

Divergência KL *forward* vs *reverse* Na literatura de ML, é comum chamar $D_{KL}(q\|p)$ de divergência KL reversa. Conversamente, $D_{KL}(p\|q)$ é conhecida como a divergência forward. Empiricamente, é bem estabelecido que minimizar a divergência reversa costuma resultar em resultados que focam em alguma(s) modas. Por outro lado, minimizar a divergência forward costuma promover aproximações que cobrem melhor o suporte de p. A figura abaixo ilustra esse fenômeno com $p(\theta) = \frac{1}{2}\mathcal{N}(\theta|3, 1) + \frac{1}{2}\mathcal{N}(\theta| - 3, (\frac{1}{2})^2)$ e q sendo Gaussiana univariada.



Vale ressaltar, que calcular o ELBO para $D_{\text{KL}}(p\|q)$ envolve calcular esperanças sob a distribuição p, o que pode ser desafiador quando p é uma posteriori.

5.3 Extensão para problemas multiclasse: classificador *softmax*

Nesse capítulo, nos focamos em problemas de classificação binária, em que $|\mathcal{Y}| = 2$. No entanto, é fácil generalizar as técnicas que discutimos para problemas multi-classe (i.e., $|\mathcal{Y}| > 2$). Para tal, basta substituir o nosso modelo observacional Bernoulli por uma distribuição categórica. Lembre que a Bernoulli é parametrizada por um parâmetro escalar que dita a probabilidade de cada classe. No caso da categórica, precisamos de um vetor de probabilidades, i.e., um vetor $\boldsymbol{r}$ de tamanho $L = |\mathcal{Y}|$, em que cada entrada r_l denota a probabilidade de classe l. Naturalmente, todas as entradas de $\boldsymbol{r}$ devem ser não-negativas e $\sum_{l=1}^{L} = 1$. Resta-nos, então, expressar $\boldsymbol{r}$ como uma função de $\boldsymbol{x}$. Para tal, podemos generalizar nosso o procedimento que usamos para regressão logística.

Primeiro, calculamos um vetor de logits $\boldsymbol{z}$, desta vez usando uma transformação linear para cada uma das L classes:

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{x}^\intercal \boldsymbol{\theta}^{(1)} \\ \boldsymbol{x}^\intercal \boldsymbol{\theta}^{(2)} \\ \vdots \\ \boldsymbol{x}^\intercal \boldsymbol{\theta}^{(L)} \end{bmatrix},$$

Finalmente, aplicamos a função Softmax para transformar $\boldsymbol{z}$ em um vetor de probabilidades e obter $\boldsymbol{r}$, que é dado por:

$$\boldsymbol{r} = \text{Softmax}(\boldsymbol{z}) = \frac{e^{\boldsymbol{z}}}{\sum_{l=1}^{L} e_l^z},$$

onde $e^{\boldsymbol{z}}$ é um abuso de notação denotando exponenciar cada uma das entradas de $\boldsymbol{z}$.

Bibliografia

Ahn, K., Yun, C. & Sra, S. (2020). SGD with shuffling: optimal rates without component convexity and large epoch requirements, Em *Advances in Neural Information Processing Systems (NeurIPS)*. (Ver página 11).

Atkeson, C. G., Moore, A. W. & Schaal, S. (1997). Locally Weighted Learning. *Artificial Intelligence Review*, *11*(1–5) (ver página 19).

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press. (Ver página 21).

Broomhead, D. & Lowe, D. (1988). Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems*, *2*, 321–355 (ver página 28).

Cover, T. & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27 (ver página 18).

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211 (ver página 50).

Lloyd, S. P. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, *28*, 129–137 (ver página 28).

Manning, C. D. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press. (Ver página 20).

McCulloch, W. S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*(4), 115–133 (ver páginas 5, 44).

Micchelli, C. A. (1986). Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constr. Approx*, *2*, 11–22 (ver página 28).

Nguyen, L., Tran-Dinh, Q., Phan, D., Nguyen, P. & van Djik, M. (2021). A Unified Convergence Analysis for Shuffling-Type Gradient Methods. *Journal of machine learning research (JMLR)* (ver página 11).

Rosenblatt, F. (1957). *The perceptron: A perceiving and recognizing automaton (Project PARA)* (relatório técnico). Cornell University (Technical Report 85-460-1). (Ver página 45).

Safran, I. & Shamir, O. (2021). Random Shuffling Beats SGD Only After Many Epochs on Ill-Conditioned Problems, Em *Advances in Neural Information Processing Systems (NeurIPS)*. (Ver página 11).

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, *49*, 433–460 (ver página 5).

Weinberger, K. Q., Blitzer, J. & Lawrence, S. K. (2006). Distance Metric Learning for Large Margin Nearest Neighbor Classification (Y. Weiss, B. Schölkopf & J. C. Platt, Editores). Em Y. Weiss, B. Schölkopf & J. C. Platt (Editores), *Advances in Neural Information Processing Systems 18*. MIT Press. (Ver página 20).