1 Análisis de la función de verosimilitud para clasificación.

Esta función de verosomilitud se toma del trabajo (1). Allí se define una variable aleatoria binaria $\lambda_n^r \in \{0,1\}$, la cual representa la confiabilidad del anotador r-ésimo al etiquetar la muestra n. De esta forma si $\lambda_n^r = 1$, se supone que y_n^r corresponde a la etiqueta verdadera ($y_n^r = y_n$), la cual se modela con una distribución categórica; por el contrario, si $\lambda_n^r = 0$, se supone que y_n^r es una versión corrupta de la etiqueta verdadera y_n, la cual se modela con una distribución uniforme. De esta forma, $\lambda_n^r \sim \text{Bernoulli}(\pi_r)$, donde π_r es la exactitud del anotador r-ésimo, y

$$p(\lambda_n^r|\pi_r) = \pi_r^{\lambda_n^r}(1-\pi_r)^{(1-\lambda_n^r)}. \tag{1}$$

Además,

$$p(\boldsymbol{Y}|\boldsymbol{\lambda},\boldsymbol{\theta}) = \prod_{n=1}^{N}\prod_{r\in R_n}\left(\prod_{k=1}^{K}\zeta_{k,n}^{\delta(y_n^r,k)}\right)^{\lambda_n^r}\left(\frac{1}{K}\right)^{(1-\lambda_n^r)}, \tag{2}$$

donde $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_R]^\top \in \{0,1\}^{NR}$, $\boldsymbol{\lambda}_r = [\lambda_1^r,\ldots,\lambda_N^r] \in \{0,1\}^N$, y $\zeta_{k,n} = p(y_n^r = k|\hat{\lambda}_n^r = 1)$. Además, $\boldsymbol{\theta} = \left\{\{\pi_r\}_{r=1}^R, \{\zeta_{k,n}\}_{n=1,k=1}^{N,K}\right\}$. De esta forma,

$$p(\boldsymbol{Y},\boldsymbol{\lambda}|\boldsymbol{\theta}) = \prod_{n=1}^{N}\prod_{r\in R_n}\left(\pi_r\prod_{k=1}^{K}\zeta_{k,n}^{\delta(y_n^r,k)}\right)^{\lambda_n^r}\left(\frac{1-\pi_r}{K}\right)^{(1-\lambda_n^r)}. \tag{3}$$

Ahora se analizan las opciones para modelar esta función de verosimilitud usando CCGPMA.

1. Marginalizar $\boldsymbol{\lambda}$ en $p(\boldsymbol{Y},\boldsymbol{\lambda}|\boldsymbol{\theta})$. De esta forma se tiene que

$$p(\boldsymbol{Y}|\boldsymbol{\theta}) = \prod_{n=1}^{N}\prod_{r\in R_n}\sum_{\lambda_n^r\in\{0,1\}}\left(\pi_r\prod_{k=1}^{K}\zeta_{k,n}^{\delta(y_n^r,k)}\right)^{\lambda_n^r}\left(\frac{1-\pi_r}{K}\right)^{(1-\lambda_n^r)}, \tag{4}$$

$$= \prod_{n=1}^{N}\prod_{r\in R_n}\left\{\pi_r\prod_{k=1}^{K}\zeta_{k,n}^{\delta(y_n^r,k)} + \frac{1-\pi_r}{K}\right\}. \tag{5}$$

   Esta opción implica modelar la exactitud del anotador como función del espacio de entrada, es decir, $\pi_n^r = h(\boldsymbol{x}_n)$. En esta opción analizo lo siguiente.
   Para calcular el ELBO, se requiere determinar el $\mathbb{E}_{q(\hat{\boldsymbol{f}})}[\log p(\boldsymbol{Y}|\boldsymbol{\theta})]$, de esta forma

$$\log p(\boldsymbol{Y}|\boldsymbol{\theta}) = \mathbb{E}_{q(\hat{\boldsymbol{f}})}\left[\log\prod_{n=1}^{N}\prod_{r\in R_n}\left\{\pi_n^r\prod_{k=1}^{K}\zeta_{k,n}^{\delta(y_n^r,k)} + \frac{1-\pi_n^r}{K}\right\}\right], \tag{6}$$

$$= \sum_{n=1}^{N}\sum_{r\in R_n}\mathbb{E}_{q(\hat{\boldsymbol{f}})}\left[\log\left\{\pi_n^r\prod_{k=1}^{K}\zeta_{k,n}^{\delta(y_n^r,k)} + \frac{1-\pi_n^r}{K}\right\}\right]. \tag{7}$$

   De acuerdo con lo anterior es necesario aproximar el valor esperado $\mathbb{E}_{q(\hat{\boldsymbol{f}})}\left[\log\left\{\pi_n^r\prod_{k=1}^{K}\zeta_{k,n}^{\delta(y_n^r,k)} + \frac{1-\pi_n^r}{K}\right\}\right]$. Si usamos Gauss-Hermite, necesitaríamos una aproximación de $J = R + K$ dimensiones, lo cual tiene un costo de almacenamiento equivalente a $\mathcal{O}(S^J)$, donde S es la cantidad de puntos usados para la aproximación (según el material suplementario de (2)). De esta forma, este tipo de aproximación no sería viable puesto que aun usando un valor pequeño para S se requiere un almacenamiento muy grande.

2. La segunda opción, es la que actualmente está implementada. A diferencia de la anterior, la idea es considerar la confiabilidad de los anotadores como un parámetro del modelo y no como una variable aleatoria Bernoulli; así se tiene

$$p(\boldsymbol{Y}|\boldsymbol{\theta}) = \prod_{n=1}^{N}\prod_{r\in R_n}\left(\prod_{k=1}^{K}\zeta_{k,n}^{\delta(y_n^r,k)}\right)^{\lambda_n^r}\left(\frac{1}{K}\right)^{(1-\lambda_n^r)},$$

donde $\boldsymbol{\theta} = \left\{ \{\lambda_n^r\}_{n=1,r=1}^{N,R}, \{\zeta_{k,n}\}_{n=1,k=1}^{N,K} \right\}$. Para modelar las confiabilidades con el CCGPMA, se implementó función de mapeo la función Sigmoidal logística. Esto repercute en que se está aproximando una variable binaria λ_n^r, con una variable $\hat{\lambda}_n^r \in \mathbb{R}$ en el intervalo $[0, 1]$. Otra alternativa (la que expuso Juan José) es usar como función de mapeo la función escalón unitario $u(t)$, la cual mapea el GP prior a unos o ceros. Yo creo que la opción con la función Sigmoidal y la opción de Juan José pueden ser equivalentes en el sentido de que uno podría considerar la función Sigmoidal logística como una aproximación suave de la función $u(t)$.

De acuerdo con lo anterior, la opción creo no es viable de acuerdo con lo expuesto, mientras que la opción dos parece ser la más viable. Enfocándonos en la opción dos, mi pregunta es si ustedes consideran correcto la decisión de usar la función Sigmoidal como función de mapeo (los resultados del paper usan esta función) y si está bien justificar el hecho de que tal función puede tomarse como una aproximación de la función $u(t)$, o si por el contrario consideran que no es correcto lo que expongo y es mejor usar el $u(t)$ como función de mapeo.

References

[1] F. Rodrigues, F. Pereira, and B. Ribeiro, "Learning from multiple annotators: Distinguishing good from random labelers," *Pattern Recognition Letters*, vol. 34, no. 12, pp. 1428–1436, 2013.

[2] P. Moreno-Muñoz, A. Artés, and M. Alvarez, "Heterogeneous multi-output Gaussian process prediction," in *Advances in neural information processing systems*, 2018, pp. 6711–6720.