Multiclass Classification

Gonzalo Hernández-Muñoz

February 20, 2019

1 How to calculate $\ln \mathcal{Z}$ for multiclass classification

NOTE: This document are just my handwritten notes on how to calculate the output for the DGP and what the equivalences between the code and equations are. Some things may not be totally accurate and there are some notation errors.

We have to calculate:

$$\begin{aligned} \mathcal{Z}_i &= \ln \mathbb{E}_{q^\setminus}[p(y_i|\mathbf{h}_i^L) \prod_{l=1}^{L} q^\setminus(\mathbf{h}_i^l|\mathbf{h}_i^{l-1})] \\ &= \int p(y_i|\mathbf{h}_i^L) \prod_{l=1}^{L} q^\setminus(\mathbf{h}_i^l|\mathbf{h}_i^{l-1}) \quad d\mathbf{h}_i^1 \ \ldots \ d\mathbf{h}_i^L \end{aligned} \tag{1}$$

At the output node we only need to calculate:

$$\ln \int p(y_i|\mathbf{h}_i^L) q^\setminus(\mathbf{h}_i^L) \quad d\mathbf{h}_i^L \tag{2}$$

Where, for multiclass classification, equals:

$$\ln \int \left[\prod_{k \neq y_i} \Theta(\mathbf{h}_{i,y_i}^L - \mathbf{h}_{i,k}^L) \right] \prod_{k=1}^{C} \mathcal{N}(\mathbf{h}_{i,k}^L|\boldsymbol{\mu}_{i,k}, \mathbf{v}_{i,k}) \quad d\mathbf{h}_{i,k=1}^L \ \ldots \ d\mathbf{h}_{i,k=C}^L \tag{3}$$

Where it is assumed that in the last layer L there are as many nodes as number of classes. The class chosen for the i-th point (y_i) will be the one in which the output value for the corresponding node is bigger.

$$y_i = \arg\max_k \ \mathbf{h}_{i,k}^L \quad k \in \{1 \ \ldots \ C\}$$

The first term (in square brackets) is one if and only if all the output values for the i-th example, i.e $\mathbf{h}_{i,k}^L$ for $k \in \{1 \ \ldots \ C\}$ except $k = y_i$. (that means, one if the point is correctly classified). Eq. 3 can be calculated as [Villacampa-Calvo, Carlos, and Daniel Hernández-Lobato. "Scalable multi-class Gaussian process classification using expectation propagation."]: (omitting subscript i except for the training example)

$$\ln \int \left[\prod_{k \neq y_i} \Phi\left(\frac{\mathbf{h}_{y_i}^L - \boldsymbol{\mu}_k}{\sqrt{\mathbf{v}_k}} \right) \right] \mathcal{N}(\mathbf{h}_{y_i}^L|\boldsymbol{\mu}_{y_i}, \mathbf{v}_{y_i}) \quad d\mathbf{h}_{y_i}^L \tag{4}$$

Where for example $\boldsymbol{\mu}_{y_i}$ means, "the mean for the output node k where $k = y_i$". And $\Phi(\cdot)$ is the CDF of a Gaussian. Now we only need to calculate a one dimensional integral (way easier).Continuing expanding 4

$$\ln \int \frac{1}{\sqrt{2\pi \mathbf{v}_{y_i}}} \exp\left[\frac{-(\mathbf{h}_{y_i}^L - \boldsymbol{\mu}_{y_i})^2}{2\mathbf{v}_{y_i}} \right] \prod_{k \neq y_i} \Phi\left(\frac{\mathbf{h}_{y_i}^L - \boldsymbol{\mu}_k}{\sqrt{\mathbf{v}_k}} \right) \quad d\mathbf{h}_{y_i}^L \tag{5}$$

(we omit the L superscript, all output values are from the last layer). We now wan to calculate the integral by using Gauss-Hermite quadrature. We need to make a variable change.

$$\mathbf{x} = \frac{\mathbf{h}_{y_i}^L - \boldsymbol{\mu}_{y_i}}{\sqrt{2\mathbf{v}_{y_i}}} \iff \mathbf{h}_{y_i}^L = \underbrace{\sqrt{2\mathbf{v}_{y_i}}\,\mathbf{x} + \boldsymbol{\mu}_{y_i}}_{\text{Called X in the code}} \tag{6}$$

$$d\mathbf{h}_{y_i}^L = \sqrt{2\mathbf{v}_{y_i}}\; d\mathbf{x} \tag{7}$$

Here, $\mathbf{x}$ are the points of the Gauss-Hermite quadrature (not related to the training points) and called **gh_x** in the code. As our method propagates S samples, we can also include the average over them ($\boldsymbol{\mu}_{y_i}$ will be of size $S, N, 1$ in the code).

$$\ln \frac{1}{S}\sum_{s=1}^{S} \int \frac{1}{\sqrt{\pi}\cancel{\sqrt{2\mathbf{v}_{y_i}}}} \exp[x^{-2}] \prod_{k\neq y_i} \Phi\left(\underbrace{\frac{\sqrt{2\mathbf{v}_{y_i}}\,\mathbf{x} + \boldsymbol{\mu}_{y_i} - \boldsymbol{\mu}_k}{\sqrt{\mathbf{v}_k}}}_{\text{Called dist in the code}^1}\right) \cancel{\sqrt{2\mathbf{v}_{y_i}}}\, d\mathbf{x} \tag{8}$$

We can apply now the Gauss-Hermite quadrature and approximate the integral. (we introduce now the weights $\mathbf{w}$, called **gh_w**)

$$\ln\left[\frac{1}{S}\sum_{s=1}^{S}\left(\frac{1}{\sqrt{\pi}}\left[\sum_{w\in\mathbf{w}} w \prod_{k\neq y_i}\Phi\left(\frac{\sqrt{2\mathbf{v}_{y_i}}\,\mathbf{x} + \boldsymbol{\mu}_{y_i} - \boldsymbol{\mu}_k}{\sqrt{\mathbf{v}_k}}\right)\right]\right)\right]$$
$$= \ln\left[\frac{1}{S\sqrt{\pi}}\sum_{s=1}^{S}\left[\sum_{w\in\mathbf{w}} w \prod_{k\neq y_i}\Phi\left(\frac{\sqrt{2\mathbf{v}_{y_i}}\,\mathbf{x} + \boldsymbol{\mu}_{y_i} - \boldsymbol{\mu}_k}{\sqrt{\mathbf{v}_k}}\right)\right]\right]$$

Note that even that it is not included in the notation, $\boldsymbol{\mu}_k$, $\mathbf{v}_{y_i}$, $\mathbf{v}_k$etc all depend on the samples.

$$= \ln\left[\sum_{s=1}^{S}\left[\sum_{w\in\mathbf{w}} w \prod_{k\neq y_i}\Phi\left(\frac{\sqrt{2\mathbf{v}_{y_i}}\,\mathbf{x} + \boldsymbol{\mu}_{y_i} - \boldsymbol{\mu}_k}{\sqrt{\mathbf{v}_k}}\right)\right]\right] - \ln\sqrt{\pi} - \ln S$$

.

It can be made more robust by introducing:

$$= \ln\left[\sum_{s=1}^{S}\left[\sum_{w\in\mathbf{w}} w \exp\left\{\sum_{k\neq y_i}\ln\Phi\left(\frac{\sqrt{2\mathbf{v}_{y_i}}\,\mathbf{x} + \boldsymbol{\mu}_{y_i} - \boldsymbol{\mu}_k}{\sqrt{\mathbf{v}_k}}\right)\right\}\right]\right] - \ln\sqrt{\pi} - \ln S$$

and the sum $\sum_{w\in\mathbf{w}}$ can be calculated as a matrix product (denoted by $\cdot$).

$$= \ln\left[\sum_{s=1}^{S}\mathbf{w}\cdot\exp\left\{\sum_{k\neq y_i}\ln\Phi\left(\frac{\sqrt{2\mathbf{v}_{y_i}}\,\mathbf{x} + \boldsymbol{\mu}_{y_i} - \boldsymbol{\mu}_k}{\sqrt{\mathbf{v}_k}}\right)\right\}\right] - \ln\sqrt{\pi} - \ln S$$

2 Prediction

For prediction we have to calculate

$$\int p(y_i = c|\mathbf{h}_{i,c}^L)\prod_{l=1}^{L} q(\mathbf{h}_{i,c}^l|\mathbf{h}_{i,c}^{l-1}) \quad d\mathbf{h}_{i,c}^1 \ldots d\mathbf{h}_{i,c}^L$$

[1]Except that in the code it is calculated for all classes and in the equation it only needs to be calculated for all the classes except the one of the training point

For all classes $c \in \{1, \ldots, K\}$. Note that this procedure is the same as in the other section but using the posterior q instead of the cavity $q^{\setminus}$. However this does not change the output node as the gp nodes are the one in charge of calculating $\mathbf{h}_{i,c}^{L}$ (for the last layer) with the corresponding distribution. (We do have to calculate each of the probabilities for each of the classes individually and the choose the one with highest probability). That is, for a test point $y_\star$

$$y_\star \leftarrow \arg\max_k \, p(y_\star = k) \quad k \in \{1 \ldots C\} \tag{9}$$