Implementation of Mixture Normal Density in NGBoost

Soumya Sahu

August 2020

1 Mixture Normal Density

Consider k normal distributions, where density of distribution j can be written as,

$$f_j(x;\theta_j,\sigma_j^2) = \frac{1}{\sqrt{2\pi}\sigma_j} exp(-\frac{(x-\theta_j)^2}{2\sigma_j^2}).$$

A random variable is said to have k-mixture normal density if the random variable follows jth normal distribution with probability α_j such that $\sum_{j=1}^{k} \alpha_j = 1$. So, the density of k-mixture normal can be written in the following form,

$$f(x,\boldsymbol{\theta}) = \sum_{j=1}^{k-1} \alpha_j f_j(x;\theta_j,\sigma_j^2) + (1-\sum_{j=1}^{k-1}\alpha_j) f_k(x;\theta_k,\sigma_k^2),$$

where, $\sigma_j > 0$, $0 < \alpha_j < 1$, $\sum_{j=1}^{k-1} \alpha_j < 1$ and $\boldsymbol{\theta} = (\theta_1,...,\theta_k,\sigma_1,...,\sigma_k,\alpha_1,...,\alpha_{k-1})^T$, vector of independent parameters.

2 Reparameterization

Implementation of NGBoost requires the parameter space must be $\mathbb{R}^d$, where d is dimension of parameter space. We transform σ_j and α_j in the following way,

$$\sigma_j{}^\star = log_e \sigma_j, \qquad j = 1,2,...,k \qquad\qquad \alpha_j{}^\star = log_e\left(\frac{\alpha_j}{1-\sum_{l=1}^{k-1}\alpha_l}\right), \qquad j = 1,2,...,k-1.$$

Here, $\sigma_j{}^\star \in \mathbb{R}$ while $\sigma_j = exp(\sigma_j{}^\star) > 0$ for $j = 1,2,...,k$.

Also, $\alpha_j{}^\star \in \mathbb{R}$ while $\alpha_j = \frac{exp(\alpha_j{}^\star)}{1+\sum_{l=1}^{k-1} exp(\alpha_l{}^\star)} < 1$ and most importantly $\sum_{l=1}^{k-1} \alpha_j = \frac{\sum_{l=1}^{k-1} exp(\alpha_l{}^\star)}{1+\sum_{l=1}^{k-1} exp(\alpha_l{}^\star)} < 1$.

So, after reparameterization, $\boldsymbol{\theta}$ becomes $\boldsymbol{\theta}^\star = (\theta_1,...,\theta_k,\sigma_1{}^\star,...,\sigma_k{}^\star,\alpha_1{}^\star,...,\alpha_{k-1}{}^\star)^T$.

3 Computing Gradient w.r.t $\boldsymbol{\theta}^{\star}$

Assume that we have a sample of size n, where the response variable is denoted as $x_1, x_2, ..., x_n$. Using Kullback-Leibler divergence, the score function (w.r.t $\boldsymbol{\theta}$) can be defined as,

$$S(\boldsymbol{\theta}) = -\frac{1}{n}\sum_{i=1}^{n} log_e f(x_i, \boldsymbol{\theta}),$$

where, $f(x, \boldsymbol{\theta})$ is the density of k-mixture normal density, defined in section 1. So, the gradient vector w.r.t $\boldsymbol{\theta}$ can be defined as,

$$\Delta S(\boldsymbol{\theta}) = \left(\frac{\delta S}{\delta \theta_1} \cdots \frac{\delta S}{\delta \theta_k} \frac{\delta S}{\delta \sigma_1} \cdots \frac{\delta S}{\delta \sigma_k} \frac{\delta S}{\delta \boldsymbol{\alpha}}\right)^T,$$

where, $\frac{\delta S}{\delta \boldsymbol{\alpha}} = \left(\frac{\delta S}{\delta \alpha_1} \cdots \frac{\delta S}{\delta \alpha_{k-1}}\right)^T$.

$$\frac{\delta S}{\delta \theta_j} = -\frac{1}{n}\sum_{i=1}^{n} \frac{\alpha_j}{f(x_i, \boldsymbol{\theta})} \frac{\delta f_j(x_i)}{\delta \theta_j},$$

where,

$$\frac{\delta f_j(x)}{\delta \theta_j} = \frac{\delta f_j(x; \theta_j, \sigma_j^2)}{\delta \theta_j} = \frac{1}{\sqrt{2\pi}} \frac{(x-\theta_j)}{\sigma_j^3} exp(-\frac{(x-\theta_j)^2}{2\sigma_j^2}), \qquad j = 1, 2, ..., k,$$

and

$$\frac{\delta S}{\delta \sigma_j} = -\frac{1}{n}\sum_{i=1}^{n} \frac{\alpha_j}{f(x_i, \boldsymbol{\theta})} \frac{\delta f_j(x_i)}{\delta \sigma_j},$$

where,

$$\frac{\delta f_j(x)}{\delta \sigma_j} = \frac{\delta f_j(x; \theta_j, \sigma_j^2)}{\delta \sigma_j} = \frac{1}{\sqrt{2\pi}} \frac{(x-\theta_j)^4 - \sigma_j^2}{\sigma_j^4} exp(-\frac{(x-\theta_j)^2}{2\sigma_j^2}), \qquad j = 1, 2, ..., k.$$

and

$$\frac{\delta S}{\delta \alpha_j} = -\frac{1}{n}\sum_{i=1}^{n} \frac{(f_j(x_i) - f_k(x_i))}{f(x_i, \boldsymbol{\theta})}.$$

In above equations, $\alpha_k = 1 - \sum_{l=1}^{k-1} \alpha_l$.

Now to find gradient w.r.t $\boldsymbol{\theta}^{\star}$ we need to compute $\frac{\delta S}{\delta \sigma_j^{\star}}$ and $\frac{\delta S}{\delta \boldsymbol{\alpha}^{\star}}$.

$$\frac{\delta S}{\delta \sigma_j^{\star}} = \frac{\delta S}{\delta \sigma_j} \frac{\delta \sigma_j}{\delta \sigma_j^{\star}} = \frac{\delta S}{\delta \sigma_j} \sigma_j, \qquad j = 1, 2, ..., k.$$

and

$$\frac{\delta S}{\delta \boldsymbol{\alpha}^\star} = A(\boldsymbol{\alpha})^{-1}\frac{\delta S}{\delta \boldsymbol{\alpha}},$$

where, $A(\boldsymbol{\alpha})$ is a matrix of dimension $(k-1)\times(k-1)$ with diagonal elements,

$$\frac{\delta {\alpha_j}^\star}{\delta \alpha_j} = \frac{1}{\alpha_j} + \frac{1}{1-\sum\limits_{l=1}^{k-1}\alpha_l}, \qquad j = 1, 2, ..., k-1$$

and off-diagonal elements,

$$\frac{\delta {\alpha_j}^\star}{\delta \alpha_m} = \frac{1}{1-\sum\limits_{l=1}^{k-1}\alpha_l}, \qquad j = 1, 2, ..., k-1, \quad m \neq j.$$

So, $A(\boldsymbol{\alpha}) = diag\{\frac{1}{\alpha_1}, \frac{1}{\alpha_2}, ..., \frac{1}{\alpha_{k-1}}\} + \frac{1}{1-\sum_{l=1}^{k-1}\alpha_l}\mathbf{1}\mathbf{1}^T$, where $\mathbf{1}$ is unit vector of length $k-1$.

So, the gradient vector w.r.t $\boldsymbol{\theta}^\star$ can be written as,

$$\Delta S(\boldsymbol{\theta}^\star) = \left(\frac{\delta S}{\delta \theta_1}\cdots\frac{\delta S}{\delta \theta_k}\frac{\delta S}{\delta {\sigma_1}^\star}\cdots\frac{\delta S}{\delta {\sigma_k}^\star}\frac{\delta S}{\delta \boldsymbol{\alpha}^\star}\right)^T.$$

We have derived $S(\boldsymbol{\theta}^\star)$ as a function of $\boldsymbol{\theta}$ but we can make it a function of $\boldsymbol{\theta}^\star$ by replacing σ_j and α_j as a function of ${\sigma_j}^\star$ and ${\alpha_j}^\star$ respectively using the derivations in section 2.

4 Computing Information Function

Information function w.r.t $\boldsymbol{\theta}^\star$ can be defined as,

$$I(\boldsymbol{\theta}^\star) = E[(\Delta S(\boldsymbol{\theta}^\star; y))(\Delta S(\boldsymbol{\theta}^\star; y))^T],$$

where, $\Delta S(\boldsymbol{\theta}^\star; y)$ is the gradient based on a single sample y from a k-mixture normal distribution.

Under regularity conditions (which are satisfied by mixture normal) $E(.)$ can be approximated by sample average. Let, $y_1, y_2, ..., y_N$ be N independent samples generated from k-mixture normal distribution. Then $I(\boldsymbol{\theta}^\star)$ can be approximated by,

$$I(\hat{\boldsymbol{\theta}}^\star) = \frac{1}{N}\sum_{i=1}^{N}(\Delta S(\boldsymbol{\theta}^\star; y_i))(\Delta S(\boldsymbol{\theta}^\star; y_i))^T.$$

I suggest N must be as large as 10000 for good approximation.

How to generate a sample of size N from k-mixture normal distribution
Draw a random number r from 1 to k, where probability of drawing number i is α_i $(i = 1, 2, ..., k)$

and generate a sample from r-th normal distribution ($N(\mu_r, \sigma_r^2)$). Replicate this N times to get N independent samples. Unfortunately, this process may be time consuming. So, we can use exchangeability property of iid samples. Generate $[N\alpha_i]$ many samples from $N(\mu_i, \sigma_i^2)$ for $i = 1, 2, ..., k-1$ and generate rest from $N(\mu_k, \sigma_k^2)$ and combine them to get total of N samples generated from k-mixture normal distribution.

5 Parameter Initialization in NGBoost Algorithm

In NGBoost algorithm, initial value of $\boldsymbol{\theta}^\star$ can be found as $argmin_{\boldsymbol{\theta}^\star} S(\boldsymbol{\theta}^\star)$ which is same as the maximum likelihood estimate of $\boldsymbol{\theta}^\star$ based on the assumption that marginal distribution of response variable x is also a mixture of normal distributions. We can use Fisher Scoring algorithm to solve this problem as we have calculated the information matrix.
Let, $\boldsymbol{\theta}^{\star(0)}$ be the maximum likelihood estimate. So, we use the following iterative step,

$$\boldsymbol{\theta}^{\star(0)}_{m+1} = \boldsymbol{\theta}^{\star(0)}_{m} - I(\boldsymbol{\theta}^{\star(0)}_{m})^{-1} \Delta S(\boldsymbol{\theta}^{\star(0)}_{m})$$

until $||\boldsymbol{\theta}^{\star 0}_{m+1} - \boldsymbol{\theta}^{\star 0}_{m}||_1 < \epsilon$. $||.||_1$ is L_1 norm and ϵ can be set to any small number like 10^{-5}.

Now an interesting question can be what is a good choice of starting value of the Fisher Scoring algorithm, i.e. what is $\boldsymbol{\theta}^{\star(0)}_{0}$. We can perform K-means clustering where we fix the number of clusters as k and sample mean, sample standard deviation from each cluster can be taken as μ, σ values in $\boldsymbol{\theta}^{\star(0)}_{0}$. We can take sample proportion of each cluster as α values in $\boldsymbol{\theta}^{\star(0)}_{0}$.