Estimating the impact of features via perturbed predictions

Christian Blume

March 26, 2020

This report presents a simple method for estimating the statistical impact of features on a supervised statistical model. It is assumed that the model has been trained. The impact is based on the mean variation of the difference between perturbed and original predictions. The impact is agnostic to the actual model being used.

Definition

The *original prediction* $\mathbf{y} \in \mathbb{R}^m$ is defined as

$$\mathbf{y} = f(\mathbf{X}) \tag{1}$$

where $f : \mathbb{R}^{m \times n} \to \mathbb{R}^m$ is a trained model, m the number of samples, and n the number of features. $\mathbf{X} \in \mathbb{R}^{m \times n}$ denotes the matrix of features such that $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ where $\mathbf{x}_k \in \mathbb{R}^m$ is the kth feature vector.

Definition

The *perturbed prediction* $\mathbf{y}_{\{k,q\}} \in \mathbb{R}^m$ is defined as

$$\mathbf{y}_{\{k,q\}} = f(\mathbf{X}_{\{k,q\}}) \tag{2}$$

where $\mathbf{X}_{\{k,q\}}$ denotes the altered feature matrix with the kth feature (column vector) held constant at a quantile q. The quantile is a representative value from the distribution of $\mathbf{x}_k$ such as the median.

Definition

The *impact* $I_{\{k,q\}}$ of feature k at quantile q is defined as

$$I_{\{k,q\}} = \sigma(\mathbf{y} - \mathbf{y}_{\{k,q\}}) \,/\, \sigma(\mathbf{x}_k) \tag{3}$$

where $\sigma(\cdot)$ is the unbiased standard deviation. $\mathbf{y} \in \mathbb{R}^m$ denotes the original prediction over m samples, $\mathbf{y}_{\{k,q\}} \in \mathbb{R}^m$ the perturbed prediction when feature k is fixed at quantile q, and $\mathbf{x}_k \in \mathbb{R}^m$ the kth feature vector.

Definition

The *averaged impact* I_k of feature k is defined as

$$I_k = \frac{1}{Q} \sum_{\forall q} I_{\{k,q\}} \tag{4}$$

where Q denotes the number of quantiles. I_k can now be normalized such that $\sum_{k=1}^{n} I_k = 1$.

Impact on a linear model

If f is linear in its model parameters $\beta \in \mathbb{R}^n$ then

$$\mathbf{y} = f(\mathbf{X}) = \mathbf{X}\beta \tag{5}$$

where $\mathbf{y} \in \mathbb{R}^m$ is the model response and $\mathbf{X} \in \mathbb{R}^{m \times n}$ the matrix of features.

Substituting Eq. 5 into Eq. 3 we receive

$$\begin{aligned} I_{\{k,q\}} &= \sigma(\mathbf{X}\beta - \mathbf{X}_{\{k,q\}}\beta) \;/\; \sigma(\mathbf{x}_k) \\ &= \sigma((\mathbf{x}_k - \mathbf{x}_{\{k,q\}})\beta_k) \;/\; \sigma(\mathbf{x}_k) \\ &= |\beta_k|\; \sigma(\mathbf{x}_k - \mathbf{x}_{\{k,q\}}) \;/\; \sigma(\mathbf{x}_k) \end{aligned} \tag{6}$$

Since $\mathbf{x}_{\{k,q\}}$ is constant for all m we receive the quantile independent estimate

$$I_k = |\beta_k| \tag{7}$$

meaning that the impact of feature k on a linear model is exactly equivalent to the absolute value of the corresponding model coefficient. This equivalence is the motivation for defining the impact as in Eq. 3.

Choice of quantiles

Quantiles are representative values of a distribution and are therefore a reasonable choice of perturbing features. If the corresponding probabilities of these quantiles are equidistant then the quantiles are bordering regions of equal area in the feature distribution. In practice, it is recommended to not use the quantiles directly but the values that are closest to the quantiles. This ensures only values are used that are actually part of the features, particularly important for distributions with multiple peaks (e.g. categorical features).

Implementation

This algorithm has been implemented as a Python package called `featureimpact` and is available on GitHub and PyPI:
`https://github.com/bloomen/featureimpact`
`https://pypi.org/project/featureimpact`