Neighborhood Component Analysis (NCA) Feature Selection

Neighborhood component analysis (NCA) is a non-parametric and embedded method for selecting features with the goal of maximizing prediction accuracy of regression and classification algorithms. The Statistics and Machine Learning Toolbox™ functions `fscnca` and `fsrnca` perform NCA feature selection with regularization to learn feature weights for minimization of an objective function that measures the average leave-one-out classification or regression loss over the training data.

NCA Feature Selection for Classification

Consider a multi-class classification problem with a training set containing n observations:

$$S = \left\{ \left(x_i, y_i \right), i = 1, 2, \dots, n \right\},$$

where $x_i \in \mathbb{R}^p$ are the feature vectors, $y_i \in \left\{1, 2, \dots, c\right\}$ are the class labels, and c is the number of classes. The aim is to learn a classifier $f : \mathbb{R}^p \rightarrow \left\{1, 2, \dots, c\right\}$ that accepts a feature vector and makes a prediction $f(x)$ for the true label y of x .

Consider a randomized classifier that:

- Randomly picks a point, $\mathrm{Ref}(x)$, from S as the ‘reference point' for x
- Labels x using the label of the reference point $\mathrm{Ref}(x)$.

This scheme is similar to that of a 1-NN classifier where the reference point is chosen to be the nearest neighbor of the new point x. In NCA, the reference point is chosen randomly and all points in S have some probability of being selected as the reference point. The probability $P\left(\mathrm{Ref}(x) = x_j | S\right)$ that point x_j is picked from S as the reference point for x is higher if x_j is closer to x as measured by the distance function d_w, where

$$d_w(x_i, x_j) = \sum_{r=1}^{p} w_r^2 |x_{ir} - x_{jr}|,$$

and w_r are the feature weights. Assume that

$$P\left(\mathrm{Ref}(x) = x_j | S\right) \propto k\left(d_w(x, x_j)\right),$$

where k is some kernel or a similarity function that assumes large values when $d_w(x, x_j)$ is small. Suppose it is

$$k(z) = \exp\left(-\frac{z}{\sigma}\right),$$

as suggested in [1]. The reference point for x is chosen from S, so sum of $P\left(\mathrm{Ref}(x) = x_j | S\right)$ for all j must be equal to 1. Therefore, it is possible to write

$$P\left(\mathrm{Ref}(x) = x_j | S\right) = \frac{k\left(d_w(x, x_j)\right)}{\sum_{j=1}^{n} k\left(d_w(x, x_j)\right)}.$$

Now consider the leave-one-out application of this randomized classifier, that is, predicting the label of x_i using the data in S^{-i}, the training set S excluding the point $\left(x_i, y_i\right)$. The probability that point x_j is picked as the reference point for x_i is

$$p_{ij} = P\left(\mathrm{Ref}(x_i) = x_j | S^{-i}\right) = \frac{k\left(d_w(x_i, x_j)\right)}{\sum_{j=1, j \neq i}^{N} k\left(d_w(x_i, x_j)\right)}.$$

The average leave-one-out probability of correct classification is the probability p_i that the randomized classifier correctly classifies observation i using S^{-i}.

$$p_i = \sum_{j=1, j \neq i}^{N} \text{probability that } x_j \text{ is the reference point for } x_i \times \mathrm{I}\left(\text{label of } j \text{ matches } i\right)$$

where $\mathrm{I}(true) = 1, \quad \mathrm{I}(false) = 0$. Then,

$$p_i = \sum_{j=1, j \neq i}^{N} p_{ij} y_{ij},$$

where

$$y_{ij} = \begin{cases} 1, & \text{if } y_i = y_j \\ 0, & \text{otherwise} \end{cases}.$$

The average leave-one-out probability of correct classification using the randomized classifier can be written as

$$F(w) = \sum_{i=1}^{N} p_i.$$

The right hand side of $F(w)$ depends on the weight vector w. The goal of neighborhood component analysis is to maximize $F(w)$ with respect to w. `fscnca` uses the regularized objective function as introduced in [1].

$$\begin{aligned} F(w) &= \sum_{i=1}^{N} p_i - \lambda \sum_{r=1}^{p} w_r^2 \\ &= \sum_{i=1}^{N} \underbrace{\sum_{j=1, j \neq i}^{N} p_{ij} y_{ij} - \lambda \sum_{r=1}^{p} w_r^2}_{F_i(w)}, \\ &= \sum_{i=1}^{N} F_i(w) \end{aligned}$$

where λ is the regularization parameter.

The regularization term drives many of the weights in w to 0. After choosing the kernel parameter σ in p_{ij} as 1, finding the weight vector w can be expressed as the following minimization problem for given λ.

$$\begin{aligned} \hat{w} = \underbrace{\text{argmin}}_{w} f(w) &= \sum_{i=1}^{N} a_i f_i(w), \\ f(w) &= -F(w), \\ f_i(w) &= -F_i(w). \end{aligned}$$

So, it finds the weights that minimize the classification error. You can specify a custom loss function using the `LossFunction` name-value pair argument in the call to `fscnca`.

NCA Feature Selection for Regression

The `fsrnca` function performs NCA feature selection modified for regression. Given n observations

$$S = \{(x_i, y_i), i = 1, 2, \ldots, n\},$$

the only difference from the classification problem is that the response values $y_i \in \mathbb{R}$ are continuous. In this case, the aim is to predict the response y given the training set S.

Consider a randomized regression model that:

- Randomly picks a point ($\text{Ref}(x)$) from Sas the ‘reference point' for x
- Sets the response value at x equal to the response value of the reference point $\text{Ref}(x)$.

Again, the probability $P(\text{Ref}(x) = x_j|S)$ that point x_j is picked from S as the reference point for x is

$$P(\text{Ref}(x) = x_j|S) = \frac{k(d_w(x, x_j))}{\sum_{j=1}^{n} k(d_w(x, x_j))}.$$

Now consider the leave-one-out application of this randomized regression model, that is, predicting the response for x_i using the data in S^{-i}, the training set S excluding the point (x_i, y_i). The probability that point x_j is picked as the reference point for x_i is

$$p_{ij} = P(\text{Ref}(x_i) = x_j|S^{-i}) = \frac{k(d_w(x_i, x_j))}{\sum_{j=1, j \neq i}^{N} k(d_w(x_i, x_j))}.$$

Let $\hat{y}_i$ be the response value the randomized regression model predicts and y_i be the actual response for x_i. And let $l : \mathbb{R}^2 \to \mathbb{R}$ be a loss function that measures the disagreement between $\hat{y}_i$ and y_i. Then, the average value of $l(y_i, \hat{y}_i)$ is

$$\begin{aligned} l_i = E\left(l(y_i, \hat{y}_i)|S^{-i}\right) &= \sum_{j=1, j \neq i}^{N} \text{probability that } x_j \text{ is the reference point for } x_i \times l(y_i, y_j) \\ &= \sum_{j=1, j \neq i}^{N} p_{ij} l(y_i, y_j) \end{aligned}$$

After adding the regularization term, the objective function for minimization is:

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} l_i + \lambda \sum_{r=1}^{p} w_r^2.$$

The default loss function $l(y_i, y_j)$ for nca for regression is mean absolute deviation, but you can specify other loss functions, including a custom one, using the `LossFunction` name-value pair argument in the call to `fsrnca`.

Impact of Standardization

The regularization term derives the weights of irrelevant predictors to zero. In the objective functions for NCA for classification or regression, there is only one regularization parameter λ for all weights. This fact requires the magnitudes of the weights to be comparable to each other. When the feature vectors x_i in S are in different scales, this might result in weights that are in different scales and not meaningful. To avoid this situation, standardize the predictors to have zero mean and unit standard deviation before applying NCA. You can standardize the predictors using the `'Standardize',true` name-value pair argument in the call to `fscnca` or `fsrnca`.

Choosing the Regularization Parameter Value

It is usually necessary to select a value of the regularization parameter by calculating the accuracy of the randomized NCA classifier or regression model on an independent test set. If you use cross-validation instead of a single test set, select the λ value that minimizes the average loss across the cross-validation folds. For examples, see Tune Regularization Parameter to Detect Features Using NCA for Classification and Tune Regularization Parameter in NCA for Regression.

References

[1] Yang, W., K. Wang, W. Zuo. "Neighborhood Component Feature Selection for High-Dimensional Data." Journal of Computers. Vol. 7, Number 1, January, 2012.

See Also

`fscnca` | `fsrnca`