kdml package

John R. J. Thompson & Jesse S. Ghashti

Introduction

The package `kdml` is a package which calculates the pairwise distances between mixed-type observations consisting of numeric (continuous), factor (nominal), and ordered factor (ordinal) variables. This kernel metric learning methodology learns the bandwidths associated with each kernel function for each variable type and returns a distance matrix that can be utilized in any distance-based clustering algorithm.

We define a kernel similarity between two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ two different way based on two papers. From Ghashti, J. S. and Thompson, J. R. J. (2023), the kernel distance summation `dkps` similarity is given by

$$s_{\text{dkps}}(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\lambda}) = \prod_{k=1}^{p_c} \frac{1}{\lambda_k} K\left(\frac{x_{i,k} - x_{j,k}}{\lambda_k}\right) + \sum_{k=p_c+1}^{p_c+p_u} L(x_{i,k}, x_{j,k}, \lambda_k) + \sum_{k=p_c+p_u+1}^{p} l(x_{i,k}, x_{j,k}, \lambda_k), \tag{1}$$

and from Ghashti, J. S. (2024), the kernel summation similarity distance `kss` is given by

$$s_{\text{kss}}(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\lambda}) = \sum_{k=1}^{p_c} K\left(\frac{x_{i,k} - x_{j,k}}{\lambda_k}\right) + \sum_{k=p_c+1}^{p_c+p_u} L(x_{i,k}, x_{j,k}, \lambda_k) + \sum_{k=p_c+p_u+1}^{p} l(x_{i,k}, x_{j,k}, \lambda_k). \tag{2}$$

For both Equations (1) and (2), $K(\cdot)$, $L(\cdot)$, and $\ell(\cdot)$ are kernel functions for continuous, nominal (factor), and ordinal (ordered factor) variables, respectively. The data frame consists of p-many variables, where $p = p_c + p_u + p_o$ indicating the total of the continuous, nominal, and ordinal variables, respectively. $\boldsymbol{\lambda}$ is a vector of length p containing variable-specific bandwidth values from each kernel functions.

Phillips, J. M., and Venkatasubramanian, S. (2011) discuss how to induce distance on similarity functions. Using either similarity function from Equations (1) or (2), the kernel distance between two observations $\mathbf{x}_i$ and $\mathbf{x}_j$ is given by

$$d^2(\mathbf{x}_i, \mathbf{x}_j \mid \boldsymbol{\lambda}) = s(\mathbf{x}_i, \mathbf{x}_i \mid \boldsymbol{\lambda}) + s(\mathbf{x}_j, \mathbf{x}_j \mid \boldsymbol{\lambda}) - 2s(\mathbf{x}_i, \mathbf{x}_j \mid \boldsymbol{\lambda}). \tag{3}$$

These two distance calculations can be called in `R` with the package `kdml` using functions `dkps` for Equation (1) and `dkss` for Equation (2), both of which are used in the Equation (3) for pairwise distance calculations.

The vector of bandwidths $\boldsymbol{\lambda}$ may user-input numeric vectors of length p, for which the bandwidths for each variable type is within the prespecified ranges. For continuous variables $\lambda > 0$, ordinal variables $\lambda \in [0, 1]$, and nominal variables is kernel specific. For example, $\lambda \in [0, 1]$ for the 'u_aitken' kernel and $\lambda \in [0, (c-1)/c]$ for 'u_aitchisonaitken', where c is the number of unique values for a specific nominal variable. For an overview of kernel functions, we refer the reader to Aitchison, J. and Aitken C.G.G. (1976), Cameron, A. and Trivedi, P. (2005), Härdle, W. et al. (2004), Li, Q. and Racine, J.S, (2007), Li, Q. and Racine, J.S, (2003), Silverman, B.W. (1986), Titterington, D.M. and Bowman, A.W. (1985), and Wang, M.C. and van Ryzin, J. (1981).

Installing

You can install the released version of kdml from Github with:

```
library(devtools)
install_github("jrjthompson/R-package-kdml")
library(kdml)
```

Bandwidth Selection

The maximization procedure for bandwidth specification is based on the objective

$$\underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \log \left(\frac{1}{(n-1)} \sum_{\substack{j=1 \\ j \neq i}}^{n} s_{\boldsymbol{\lambda}}(\mathbf{x}_i, \mathbf{x}_j) \right) \right\}, \tag{4}$$

where $s_{\boldsymbol{\lambda}}(\cdot)$ is either of the similarity functions from Equations (1) and (2). This bandwidth optimization technique, called maximum-similarity cross-validation (MSCV) and be invoked directly within the distance calculation by setting the argument `bw = "mscv"` in the functions `dkps` and `dkss`.

Users also have the option of setting the argument `bw = "np"` to specify bandwidth selection using maximum-likelihood cross-validation from the high optimized package `np` (Racine, J. S., and Hayfield, T., 2023).

Sample Usage

We simulate a mix of continuous (x_1, x_4), nominal (x_2, x_3), and ordinal data (x_5, x_6) an store in a data frame as follows

```
df <- data.frame(
  x1 = runif(100, 0, 100),
  x2 = factor(sample(c("A", "B", "C"), 100, replace = TRUE)),
  x3 = factor(sample(c("A", "B", "C"), 100, replace = TRUE)),
  x4 = rnorm(100, 10, 3),
  x5 = ordered(sample(c("Low", "Medium", "High"), 100, replace = TRUE),
               levels = c("Low", "Medium", "High")),
  x6 = ordered(sample(c("Low", "Medium", "High"), 100, replace = TRUE),
               levels = c("Low", "Medium", "High"))
)
```

Minimal implementation requires only the data frame, where the functions default the kernel functions, and the bandwidth specification method to 'mscv'.

```
# DKPS distance (Equation 1 and 3)
dis_dkps <- dkps(df = df)

# DKSS distance (Equations 2 and 3)
dis_kdss <- kdss(df = df)
```

Using the maximum-likelihood cross-validation technique from package `np`.

```
# DKPS distance (Equation 1 and 3)
dis_dkps_np <- dkps(df = df, bw = "np")

# DKSS distance (Equations 2 and 3)
dis_kdss_np <- kdss(df = df, bw = "np")
```

Users also have many kernel functions available them, which are listed in the additional arguments below. Some of the kernel functions are not available with `np`, and kernels used for the bandwidth specification technique should be the same used for the distance calculation

```
dis_dkps_custom_kernels <- dkss(df = df, bw = "mscv",
    cFUN = "c_epanechnikov", uFUN = "u_aitken", oFUN = "o_habbema")
```

If users require only the bandwidths specified from the MSCV formula in Equation (4), and not the pairwise distance matrix obtained from `dkps` or `dkss`, they may do so with the following function calls:

```
# MSCV bandwidth specification using the similarity function in Equation (1)
mscv.dkps(df, nstart = NULL, ckernel = "c_gaussian",
           ukernel = "u_aitken", okernel = "o_wangvanryzin", verbose = TRUE)

# MSCV bandwidth specification using the similarity function in Equation (2)
mscv.dkss(df, nstart = NULL, ckernel = "c_gaussian",
          ukernel = "u_aitken", okernel = "o_wangvanryzin", verbose = TRUE)
```

Arguments for the `dkps` and `dkss` functions:

- `df`: a p-variate data frame for which the pairwise distances between observations will be calculated. The data types may be continuous, nominal (unordered factors), ordinal (ordered factors), or any combination thereof. Columns of `df` should be of appropriate variable type prior to running the function.
- `bw`: a bandwidth specification method. This can be set as a vector of p-many bandwidths, with each element i corresponding to the bandwidth for column i in `df`. Alternatively, one of two character strings may be inputted for bandwidth selection methods. `mscv` specifies maximum-similarity cross-validation, and `np` specifies likelihood-cross validation which is calculated via `npudensbw` in package `np`. Defaults to `mscv`.
- `cFUN`: character string specifying the continuous kernel function. Options include `c_gaussian`, `c_epanechnikov`, `c_uniform`, `c_triangle`, `c_biweight`, `c_triweight`, `c_tricube`, `c_cosine`, `c_logistic`, `c_sigmoid`, and `c_silverman`. Note that if using `np` for `bw` selection above, continuous kernel types are restricted to either `c_gaussian`, `c_epanechnikov`, or `c_uniform`. Defaults to `c_gaussian`.
- `uFUN`: character string specifying the nominal kernel function for unordered factors. Options include `u_aitken` and `u_aitchisonaitken`. Defaults to `u_aitken`.
- `oFUN`: character string specifying the ordinal kernel function for ordered factors. Options include `o_aitken`, `o_aitchisonaitken`, `o_habbema`, `o_wangvanryzin`, and `o_liracine`. Note that if using `np` for `bw` selection above, ordinal kernel types are restricted to either `o_wangvanryzin` or `o_liracine`. Defaults to `o_wangvanryzin`.
- `stan`: a logical value which specifies whether to scale the resulting distance matrix between 0 and 1 using min-max normalization. If set to `FALSE`, there is no normalization. Defaults to `TRUE`.

The resulting outputs are the following:

- `distances`: an $n \times n$ numeric matrix containing pairwise distances between observations.
- `bandwidths`: a p-variate vector of bandwidth values returned based on the bw bandwidth specification method, sorted by variable type.

References

[1] Aitchison, J. and C.G.G. Aitken (1976), "Multivariate binary discrimination by the kernel method," Biometrika, 63, 413-420.

[2] Cameron, A. and P. Trivedi (2005), "Microeconometrics: Methods and Applications", Cambridge University Press.

[3] Ghashti, J.S. (2024), Similarity Maximization and Shrinkage Approach in Kernel Metric Learning for Clustering Mixed-type Data (T), University of British Columbia.

[4] Ghashti, J.S. and J.R.J Thompson (2023), "Mixed-type Distance Shrinkage and Selection for Clustering via Kernel Metric Learning." arXiv preprint arXiv:2306.01890.

[5] Härdle, W., and M. Müller and S. Sperlich and A. Werwatz (2004), Nonparametric and Semiparametric Models, (Vol. 1). Berlin: Springer.

[6] Hayfield, T. and J.S. Racine (2008). Nonparametric Econometrics: The np Package. Journal of Statistical Software 27(5).

[7] Li, Q. and J.S. Racine (2007), Nonparametric Econometrics: Theory and Practice, Princeton University Press.

[8] Li, Q. and J.S. Racine (2003), "Nonparametric estimation of distributions with categorical and continuous data," Journal of Multivariate Analysis, 86, 266-292.

[9] Silverman, B.W. (1986), Density Estimation, London: Chapman and Hall.

[10] Titterington, D.M. and A.W. Bowman (1985), "A comparative study of smoothing procedures for ordered categorical data", Journal of Statistical Computation and Simulation, 21(3-4), 291-312.

[11] Wang, M.C. and J. van Ryzin (1981), "A class of smooth estimators for discrete distributions," Biometrika, 68, 301-309.