index

For data $X_{n \times p}$, the projection basis A_{p*d} gives the projected data, $y = XA$. Projection pursuit finds the projection direction A that maximises the index function f:

$$\max f(XA) \quad A'A = 1.$$

For 2D projections, we can also write $y_{n \times 2} = (y_1, y_2)$, where y_1 and y_2 are the two columns of the projected data.

The following sections summarise the index functions used in the simulation:

hole index

- smooth
- formula:

$$I_{holes}(A) = \frac{1 - 1/n \sum_{i=1}^{n} \exp(-1/2 y_i y_i')}{1 - \exp(-p/2)}$$

- reference:
  - Cook and Swayne (2007) Interactive and Dynamic Graphics for Data Analysis page 30: link
  - code from the tourr package

dcor2d_2 index

- smooth
- formula:
  - define pair-wised distance: $a_{ij} = \|y_1^i - y_1^j\|$ and $b_{ij} = \|y_2^i - y_2^j\|$ for $i, j = 1, 2, \ldots, n$.

- define the column sum, row sum, and overall sum:

$$a_{i\cdot} = \sum_{l=1}^{n} a_{il}, \quad a_{\cdot j} = \sum_{k=1}^{n} a_{kj}, \quad a_{\cdot\cdot} = \sum_{k,l=1}^{n} a_{kl} \quad b_{i\cdot} = \sum_{l=1}^{n} b_{il}, \quad b_{\cdot j} = \sum_{k=1}^{n} b_{kj}, \quad b_{\cdot\cdot} = \sum_{k,l=1}^{n} b_{kl}$$

- define the distance covariance and variance:

$$\mathrm{dCov}(a_{ij}, b_{ij}) = \frac{1}{n(n-3)} \sum_{i \neq j} a_{ij} b_{ij} - \frac{2}{n(n-2)(n-3)} \sum_{i=1}^{n} a_{i\cdot} b_{i\cdot} + \frac{1}{n(n-1)(n-2)(n-3)} a_{\cdot\cdot} b_{\cdot\cdot}$$

- define the distance correlation:

$$\mathrm{dCor}(a_{ij}, b_{ij}) = \frac{\mathrm{dCov}(a_{ij}, b_{ij})}{\sqrt{\mathrm{dVar}(a_{ij}) \mathrm{dVar}(b_{ij})}}$$

- Reference:
  - Xiaoming Huo &Gábor J. Székely (2016) Fast Computing for Distance Covariance: link. The papre proves that the covariance formula above (eq 3.3 in the paper) is an unbaised estimator of the population covariance in the original formula: $\mathrm{dCov}_n^2(X, Y) := \frac{1}{n^2} \sum_{j=1}^{n} \sum_{k=1}^{n} A_{j,k} B_{j,k}$. The fast algorithm realises the new formula is in $O(n \log n)$ time.
  - Maria L. Rizzo, Gábor J. Székely (2015) Energy distance: link WIREs Computational Statistics review paper
  - Gábor J. Székely, Maria L. Rizzo, Nail K. Bakirov (2007) Measuring and testing dependence by correlation of distances: link: original paper of distance correlation

MIC/ TIC index

- smooth
- formula
  - Let $g(k, l)$ define a partition of the space (y_1, y_2) into $k \times l$ rectangles, for example, $g(2, 3)$ means dividing the data space into 2 rectangles in the y_2 direction and 3 rectangles in the y_1 direction. Let G denotes all the possible partition.
  - MIC finds the maximum mutual information over G where k and l is bounded by the grid size: $k \times l < B(n) = n^{\alpha}$ where alpha = 0.3 in our simulation

$$I_{MIC}(A) = \max_{g \in G} \frac{I(y_1, y_2|g)}{\log(\min(k^*, l^*))} = \max_{g \in G} \frac{\sum_{y1} \sum_{y2} P(y_1, y_2) \log \frac{P(y_1)}{P(y_1)P(y_2)}}{\log(\min(k^*, l^*))}$$

where k^* and l^* are the number of rectangles in the optimal partition g.

- TIC calculates the sum of mutual information over G

$$I_{TIC}(A) = \sum_{g \in G} \frac{I(y_1, y_2|g)}{\log(\min(k^*, l^*))} = \sum_{g \in G} \frac{\sum_{y1} \sum_{y2} P(y_1, y_2) \log \frac{P(y_1)}{P(y_1)P(y_2)}}{\log(\min(k^*, l^*))}$$

- Reference:
  - Reshef (2011) Detecting Novel Associations in Large Datasets: link - Figure 1 is usually to understand the general idea
  - Reshef (2016) Measuring Dependence Powerfully and Equitably: link.
- Note:
  - The original Definition 1 on page 6 of Reshef (2016) uses supremum, here I just use the maximum.
  - There are different version of MIC and we use MIC_e and TIC_e (See Sec 4 in the paper in Reshef (2016)), which uses the equicharacteristic matrix rather than the original characteristic matrix in both MIC/ TIC calculation, see page 14 Figure 1 for there difference (equicharacteristic matrix parts the space equally hence faster).
  - Section 4.3/4.4 of Reshef (2016) talks about the time complexity

loess/ spline index

- smooth
- formula:
  - let $e^{\text{model}}_{y_1 \sim y_2}$ denote the residual from model $y_1 \sim y_2$
  - for the loess index, we fit a loess model to $y_1 \sim y_2$ (use $\alpha = 0.05$ for the smoothing parameter in the loess index)

$$I_{\text{loess}}(A) = \max\left(1 - \frac{var(e^{\text{loess}}_{y_1 \sim y_2})}{var(y_1)}, 1 - \frac{var(e^{\text{loess}}_{y_2 \sim y_1})}{var(y_2)}\right)$$

- for the spline index, a spline model is fitted (use cubic regression spline with k = 15 for the diemnsion of the basis in the spline model).

$$I_{\text{spline}}(A) = \max\left(1 - \frac{var(e^{\text{spline}}_{y_1 \sim y_2})}{var(y_1)}, 1 - \frac{var(e^{\text{spline}}_{y_2 \sim y_1})}{var(y_2)}\right)$$

- reference:
  - loess: (none) I create it myself, inspired by the spline index
  - spline:
    * Ursula and Di (2020) Using tours to visually investigate properties of new projection pursuit indexes with application to problems in physics, page 1176: link
    * code from the cassowaryr page

Stringy

- non-smooth
- formula:

$$I_{stringy}(A) = \frac{\text{number of vertices with 2 edges}}{\text{number of total vertices with more than one edge}}$$

  (probably need some graph theory notation to write it mathematically)

- reference:
  - Wilkinson et al (2005) Graph-Theoretic Scagnostics: page 160: link
  - code from the cassowaryr page