Multivariate statistical dependence measure based on characteristic functions

povilas.daniusis, povilasd@neurotechnology.com

November 2021

Abstract

In this paper we propose multivariate statistical dependence measure based on the difference between joint and product of marginal characteristic functions. We discuss simulated examples, and applications for feature selection/extraction, causal inference and conduct corresponding experiments with diverse collection of multivariate data sets.

1 Introduction

Estimation of statistical dependence, both qualitatively and quantitatively, plays important role in various statistical and machine learning methods (e.g. hypothesis testing, feature selection and extraction [**?**], information bottleneck methods [**?**], cost function / reinforcement learning reward design, causal inference [**?**], among others). Therefore, earliest statistical dependence estimation ideas (e.g. conditional probability) likely share nearly-common origin with the beginning of formal statistical reasoning itself. During last two centuries ideas of correlation and (relative) entropy (including various generalizations) were proposed and became very popular in numerous applications and theoretical developments. However, with the increasing popularity of statistical machine learning, new statistical dependence estimation methods, that are robust, applicable to noisy, high-dimensional data, and can be efficiently integrated with modern machine learning methods are helpful for the development both of the theory and application.

In this study we will review some important previous dependence estimation approaches (Section 2), devoting special attention to ones based on characteristic functions (Section 2.1). Afterwards we formulate new characteristic function-based statistical dependence measure and its empirical estimator (Section 3). Section 4 is devoted to experiments with simulated and real data sets, and finalizing Section 5 concludes this article.

2 Previous work

2.1 Characteristic-function-based methods

[?]

3 Proposed Method

We will derive independence measure relying on property of characteristic functions (also known as Kac theorem [?]), that independence of two random vectors $X \in R^{d_x}$ and $Y \in R^{d_y}$ is equivalent to $\forall \alpha \in R^{d_x}, \beta \in R^{d_y}$,

$$\mathbb{E}_{X,Y} e^{i<\alpha,X>+i<\beta,Y>} = \mathbb{E}_X e^{i<\alpha,X>} \mathbb{E}_Y e^{i<\beta,Y>}, \tag{1}$$

where $i = \sqrt{-1}$, d_x and d_y are dimensions of X and Y, respectively.
This motivates the construction of a novel statistical dependence measure, which we further refer to as Kac independence measure (KacIM):

$$\kappa(X,Y) = \max_{\alpha,\beta} |\mathbb{E}_{X,Y} e^{i<\alpha,X>+i<\beta,Y>} - \mathbb{E}_X e^{i<\alpha,X>} \mathbb{E}_Y e^{i<\beta,Y>}| \tag{2}$$

It is easy to see that $0 \leq \kappa(X,Y) \leq 1$, $\kappa(X,Y) = \kappa(Y,X)$.

3.1 Estimation

Having i.i.d. data (x_j, y_j), $j = 1, 2, ..., n$ an empirical estimator of KacIM (2) is defined via corresponding empirical characteristic functions:

$$\hat{\kappa}(X,Y) = \max_{\|\alpha\|=\|\beta\|=1} |\frac{1}{n} \sum_{j=1}^{n} e^{i(<\alpha,x_j>+<\beta,y_j>)} - \frac{1}{n^2} \sum_{j=1}^{n} e^{i<\alpha,x_j>} \sum_{k=1}^{n} e^{i<\beta,y_k>}|. \tag{3}$$

Empirical estimator also admits $0 \leq \hat{\kappa}(X,Y) \leq 1$. Normalisation of parameters α and β on to unit sphere is included due to stability issues. The estimator (3) can be calculated by using Algorithm 1.

Algorithm 1 KacIM estimator computation algorithm

Require: data batch (x, y), gradient-based optimiser $GradOpt(loss)$
Calculate KacIM estimator $\hat{\kappa}(x, y)$, without maximization step (i.e. using current α, β).
Perform one maximization iteration of computed $\hat{\kappa}(x, y)$ via $\alpha, \beta \rightarrow GradOpt(\hat{\kappa}(x, y))$.

4 Experiments

Dependence measures have board area of applications. For example, regularization [?, ?], feature selection and extraction [?], information bottleneck methods

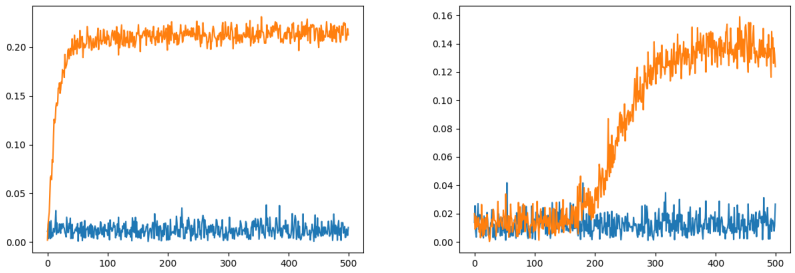

Figure 1: Dependence detection in additive (left) and multiplicative (right) noise scenario.

[**?**], causal inference [**?**], among others. Further we will conduct empirical investigation of KacIM. Starting with simple illustrative simulations, we will reformulate some key ideas in aforementioned topics for KacIM, and experimentally investigate corresponding empirical scenarios.

4.1 Generated data

We begin with simple example, which demonstrates the efficiency of KacIM for simulated multivariate data with additive and multiplicative noise.

In Figure 4.1 reflects KacIM values during iterative adaptation (500 iterations). In the case of independent data, both x_i and y_i ($d_x = 1024$, $d_y = 32$) are sampled from gaussian distribution, independently. In the case of dependent data, an additive noise (left graph) and multiplicative noise (right graph), the dependent variable is generated according to $y_i = sin(Px_i) + cos(Px_i) + \lambda\epsilon_i$ ($\lambda = 0.15$) and $y_i = (sin(Px_i) + cos(Px_i))\epsilon_i$, respectively, where P is $d_x \times d_y$ random projection matrix, $\epsilon_i \sim N(0, 1)$ and $\epsilon_i \perp x_i$.

When data is independent (blue graph), both in additive and multiplicative cases, due to independence, estimator (3) is resistant to maximization, and oscillates near zero. On the other hand, when data is not independent (orange graph), the condition of Kac theorem is violated and maximization of estimator (3) is possible.

4.2 Influence of noise level λ to KacIM estimator value

In this simulation we use the same additive noise setting as in previous paragraph, but evaluate all noise levels $\lambda \in [0.0, 2.4]$, with step 0.1. Figure 4.2 empirically shows that value of KacIM correlates with noise level, and therefore the proposed measure is able not only to detect whether independence is present, but also to quantitatively evaluate it, which enables to use KacIM to derive cost functions for vairous other learning-based algorithms.

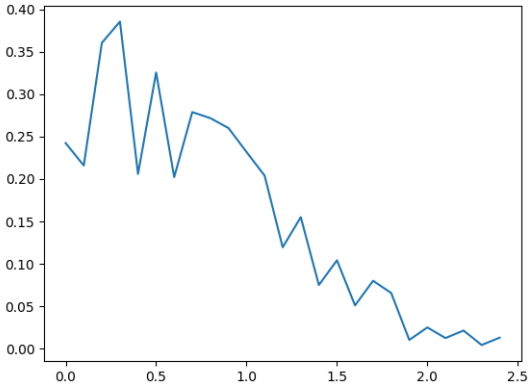

Figure 2: Noise level (x axis) vs final iteration KacIM value (y axis). KacIM values for larger noise levels saturates as in tail of graph.

4.3 Feature Extraction

We conduct linear feature extraction by seeking

$$W^* = arg \max_W \kappa(Wx, y). \tag{4}$$

Afterwards, feature extraction is conducted by $f = W^*x$ and k-nearest neighbor classification with Euclidean distance is performed, comparing unmodified inputs x and features of all possible dimensions up to d_x.

5 Discussion