Non-Parametric Jensen-Shannon Divergence

Hoang-Vu Nguyen and Jilles Vreeken

Max-Planck Institute for Informatics and Saarland University, Germany
{hnguyen,jilles}@mpi-inf.mpg.de

Abstract. Quantifying the difference between two distributions is a common problem in many machine learning and data mining tasks. What is also common in many tasks is that we only have empirical data. That is, we do not know the true distributions nor their form, and hence, before we can measure their divergence we first need to assume a distribution or perform estimation. For exploratory purposes this is unsatisfactory, as we want to explore the data, not our expectations. In this paper we study how to non-parametrically measure the divergence between two distributions. More in particular, we formalise the well-known Jensen-Shannon divergence using cumulative distribution functions. This allows us to calculate divergences directly and efficiently from data without the need for estimation. Moreover, empirical evaluation shows that our method performs very well in detecting differences between distributions, outperforming the state of the art in both statistical power and efficiency for a wide range of tasks.

1 Introduction

Measuring the difference between two distributions – their divergence – is a key element of many data analysis tasks. Let us consider a few examples. In time series analysis, for instance, to detect either changes or anomalies we need to quantify how different the data in two windows is distributed [18, 23]. In discretisation, if we want to maintain interactions, we should only merge bins when their multivariate distributions are similar [13]. In subgroup discovery, the quality of a subgroup depends on how much the distribution of its targets deviates from that of its complement data set [3, 6].

To optimally quantify the divergence of two distributions we need the actual distributions. Particularly for exploratory tasks, however, we typically only have access to empirical data. That is, we do not know the actual distribution, nor even its *form*. This is especially true for real-valued data. Although we can always make assumptions (parametric) or estimating them by kernel density estimation (KDE), these are not quite ideal in practice. For example, both parametric and KDE methods are prone to the curse of dimensionality [22]. More importantly, they restrict our analysis to the specific types of distributions or kernels used. That is, if we are not careful we are exploring our expectations about the data, not the data itself. To stay as close to the data as possible, we hence study a non-parametric divergence measure.

In particular, we propose CJS, an information-theoretic divergence measure for numerical data. We build it upon the well-known Jensen-Shannon (JS) divergence. Yet, while the latter works with probability distribution functions (pdfs), which need to be estimated, we consider *cumulative* distribution functions (cdfs) which can be obtained directly from data. CJS has many appealing properties. In a nutshell, it does not make assumptions on the distributions or their relation, it permits non-parametric computation on empirical data, and is robust against the curse of dimensionality.

Empirical evaluation on both synthetic and real-world data for a wide range of exploratory data analysis tasks including change detection, anomaly detection, discretisation, and subgroup discovery shows that CJS consistently outperforms the state of the art in both quality and efficiency.

Overall, the main contributions of this paper are as follows:

(a) a new information-theoretic divergence measure CJS,
(b) a non-parametric method for computing CJS on empirical data, and
(c) a wide range of experiments on various tasks that validate the measure.

The road map of this paper is as follows. In Section 2, we introduce the theory of CJS. In Section 3, we review related work. In Section 4 we evaluate CJS empirically. We round up with discussion in Section 5 and finally conclude in Section 6. For readability and succinctness, we postpone the proofs for the theorems to Appendix A.

2 Theory

We consider numerical data. Let X be a univariate random variable with $dom(X) \subseteq \mathbb{R}$, and let $\mathbf{X}$ be a multivariate random variable $\mathbf{X} = \{X_1, \ldots, X_m\}$, with $\mathbf{X} \subseteq \mathbb{R}^m$. Our goal is to measure the difference between two distributions $p(\mathbf{X})$ and $q(\mathbf{X})$ over the same random variable, where we have n_p and n_q data samples, respectively. We will write p and q to denote the pdfs, and say P and Q for the respective cdfs. All logarithms are to base 2, and by convention we use $0 \log 0 = 0$.

Ideally, a divergence measure gives a zero score iff $p(\mathbf{x}) = q(\mathbf{x})$ for every $\mathbf{x} \in dom(\mathbf{X})$. That is, $p(\mathbf{X}) = q(\mathbf{X})$. Second, it is often convenient if the score is symmetric. Third, it should be well-defined without any assumption on the values of $p(\mathbf{x})$ and $q(\mathbf{x})$ for $\mathbf{x} \in dom(\mathbf{X})$. That is, no assumption the relation between p and q needs to be made. Fourth, to explore the data instead of exploring our expectations, the measure should permit non-parametric computation on empirical data. Finally, as real-world data often has high dimensionality and limited observations, the measure should be robust to the curse of dimensionality.

To address each of these desired properties, we propose CJS, a new information-theoretic divergence measure. In short, CJS embraces the spirit of Kullback-Leibler (KL) and Jensen-Shannon (JS) divergences, two well-known information-theoretic divergence measures. They both have been employed widely in data mining [8, 12]. As we will show, however, in their traditional form both suffer from some drawbacks w.r.t. exploratory analysis. We will alleviate these issues with CJS.

2.1 Univariate Case

To ease presentation, let us discuss the univariate case; when $\mathbf{X}$ is a single variable.

On univariate distributions, we consider a single univariate random variable X. We start with Kullback-Leibler divergence – one of the first information-theoretic divergences proposed in statistics [9]. Conventionally, it is defined as follows.

$$\textsc{kl}(p(X) \mid\mid q(X)) = \int p(x) \log \frac{p(x)}{q(x)} dx \quad .$$

Importantly, it holds that $\textsc{kl}(p(X) \mid\mid q(X)) = 0$ iff $p(X) = q(X)$. Although KL is assymetic itself, we can easily achieve symmetry by using $\textsc{kl}(p(X) \mid\mid q(X)) + \textsc{kl}(q(X) \mid\mid p(X))$. In addition, KL does suffer from two issues, however. First, it is undefined if $q(x) = 0$ and $p(x) \neq 0$, or vice versa, for some $x \in dom(X)$. Thus, p and q have to be *absolutely continuous* w.r.t. each other for their KL score to be defined [11]. As a result, KL requires an assumption on the relationship between p and q. Second, KL works with pdfs which need parametric or KDE estimation.

Another popular information-theoretic divergence measure is the Jensen-Shannon divergence [11]. It is defined as

$$\textsc{js}(p(X) \mid\mid q(X)) = \int p(x) \log \frac{p(x)}{\frac{1}{2}p(x) + \frac{1}{2}q(x)} dx \quad .$$

As for KL, for JS we also have that $\textsc{js}(p(X) \mid\mid q(X)) = 0$ iff $p(X) = q(X)$, and we can again obtain symmetry by considering $\textsc{js}(p(X) \mid\mid q(X)) + \textsc{js}(q(X) \mid\mid p(X))$. In contrast, JS is well defined independent of the values of $p(x)$ and $q(x)$ with $x \in dom(X)$. However, it still requires us to know or estimate the pdfs.

To address this, that is, to address the computability of JS on empirical data, we propose to redefine it by replacing pdfs with cdfs. This gives us a new divergence measure, CJS, for cumulative JS divergence.

Definition 1 (Univariate CJS). *The cumulative* JS *divergence of* $p(X)$ *and* $q(X)$*, denoted* $\textsc{cjs}(p(X) \mid\mid q(X))$*, is*

$$\int P(x) \log \frac{P(x)}{\frac{1}{2}P(x) + \frac{1}{2}Q(x)} dx + \frac{1}{2 \ln 2} \int (Q(x) - P(x))\, dx \quad .$$

As we will explain shortly below, the second integral is required to make the score non-negative. Similar to KL and JS, we address symmetry by considering $\textsc{cjs}(p(X) \mid\mid q(X)) + \textsc{cjs}(q(X) \mid\mid p(X))$. Similar to JS, our measure does not make any assumption on the relation of p and q. With the following theorem we proof that CJS is indeed a divergence measure.

Theorem 1. $\textsc{cjs}(p(X) \mid\mid q(X)) \geq 0$ *with equality iff* $p(X) = q(X)$.

Proof. Applying in sequence the log-sum inequality, and the fact that $\alpha \log \frac{\alpha}{\beta} \geq \frac{1}{\ln 2}(\alpha - \beta)$ for any $\alpha, \beta > 0$, we obtain

$$\begin{aligned} \int P(x) \log \frac{P(x)}{\frac{1}{2}P(x) + \frac{1}{2}Q(x)} dx &\geq \int P(x) dx \log \frac{\int P(x) dx}{\int (\frac{1}{2}P(x) + \frac{1}{2}Q(x)) dx} \\ &\geq \frac{1}{2 \ln 2} \int (P(x) - Q(x))\, dx \quad . \end{aligned}$$

For the log-sum inequality, equality holds if and only if $\frac{P(x)}{\frac{1}{2}P(x)+\frac{1}{2}Q(x)} = \delta$ for every $x \in dom(X)$ with δ being a constant. Further, equality of the second inequality holds if and only if $\int P(x)dx = \int(\frac{1}{2}P(x) + \frac{1}{2}Q(x))dx$. Combining the two, we arrive at $\delta = 1$, i.e. $P(x) = Q(x)$ for every $x \in dom(X)$. Taking the derivatives of the two sides, we obtain the result. □

In Sec. 2.3, we will show in more detail that by considering cdfs, CJS permits non-parametric computation on empirical data. Let us now consider multivariate variables.

2.2 Multivariate Case

We now consider multivariate $\mathbf{X}$. In principle, the multivariate versions of KL and JS are obtained by replacing X with $\mathbf{X}$. We could arrive at a multivariate version of CJS in a similar way. However, if we were to do so, we would have to work with the joint distribution over *all* dimensions in $\mathbf{X}$, which would make our score prone to the curse of dimensionality. To overcome this, we build upon a factorised form of KL, as follows.

Theorem 2. $\text{KL}\left(p(\mathbf{X}) \mid\mid q(\mathbf{X})\right) =$

$$\begin{aligned} &\text{KL}(p(X_1) \mid\mid q(X_1)) + \text{KL}(p(X_2 \mid X_1) \mid\mid q(X_2 \mid X_1)) \\ &+ \ldots + \\ &\text{KL}(p(X_m \mid \mathbf{X} \setminus \{X_m\}) \mid\mid q(X_m \mid \mathbf{X} \setminus \{X_m\})) \end{aligned}$$

where

$$\begin{aligned} &\text{KL}(p(X_i \mid X_1, \ldots, X_{i-1}) \mid\mid q(X_i \mid X_1, \ldots, X_{i-1})) \\ &= \int \text{KL}(p(X_i \mid x_1, \ldots, x_{i-1}) \mid\mid q(X_i \mid x_1, \ldots, x_{i-1})) \\ &\qquad \times p(x_1, \ldots, x_{i-1}) \times dx_1 \times \ldots \times dx_{i-1} \end{aligned}$$

is named an $(i-1)$*-order conditional* KL *divergence.*

Proof. We extend the proof of Theorem 2.5.3 in [5] to the multivariate case. □

Theorem 2 states that $\text{KL}(p(\mathbf{X}) \mid\mid q(\mathbf{X}))$ is the summation of the difference between univariate (conditional) pdfs. This form of KL is less prone the curse of dimensionality thanks to the low-order conditional divergence terms. We design the multivariate version of CJS along the same lines. In particular, directly following Theorem 2 multivariate CJS is defined as.

Definition 2 (Fixed-Order CJS). $\text{CJS}(p(X_1, \ldots, X_d) \mid\mid q(X_1, \ldots, X_d))$ *is*

$$\begin{aligned} &\text{CJS}(p(X_1) \mid\mid q(X_1)) + \text{CJS}(p(X_2 \mid X_1) \mid\mid q(X_2 \mid X_1)) \\ &+ \ldots + \\ &\text{CJS}(p(X_d \mid \mathbf{X} \setminus \{X_d\}) \mid\mid q(X_d \mid \mathbf{X} \setminus \{X_d\})) \end{aligned}$$

where

$$\begin{aligned}&\textsc{cjs}(p(X_i \mid X_1, \ldots, X_{i-1}) \mid\mid q(X_i \mid X_1, \ldots, X_{i-1}))\\ &= \int \textsc{cjs}(p(X_i \mid x_1, \ldots, x_{i-1}) \mid\mid q(X_i \mid x_1, \ldots, x_{i-1}))\\ &\qquad\qquad \times p(x_1, \ldots, x_{i-1}) \times dx_1 \times \ldots \times dx_{i-1}\end{aligned}$$

is named an $(i-1)$-order conditional CJS *divergence.*

From Definition 2, one can see the analogy between multivariate CJS and the factorised form of KL. However, unlike KL, when defined as in Definition 2 CJS may be variant to how we factorise the distribution, that is, the permutation of dimensions. To circumvent this we derive a permutation-free version of CJS as follows. Let $\mathcal{F}$ be the set of bijective functions $\sigma : \{1, \ldots, m\} \to \{1, \ldots, m\}$.

Definition 3 (Order-Independent CJS). $\textsc{cjs}(p(\mathbf{X}) \mid\mid q(\mathbf{X}))$ *is*

$$\max_{\sigma \in \mathcal{F}} \sum_{i=2}^{d} \textsc{cjs}\left(p(X_{\sigma(1)}, \ldots, X_{\sigma(m)}) \mid\mid q(X_{\sigma(1)}, \ldots, X_{\sigma(m)})\right) \quad .$$

Definition 3 eliminates the dependence on any specific permutation by taking the maximum score over all permutations. Now we need to show that multivariate CJS is indeed a divergence.

Theorem 3. $\textsc{cjs}(p(\mathbf{X}) \mid\mid q(\mathbf{X})) \geq 0$ *with equality iff* $p(\mathbf{X}) = q(\mathbf{X})$.

Proof. We postpone the proof to Appendix A. □

We now know that CJS is a suitable divergence measure for multivariate distributions. To compute CJS, however, we would have to search for the optimal permutation among $m!$ permutations. When m is large, this is prohibitively costly. We tackle this by proposing $\textsc{cjs}_{pr}$, a practical version of CJS.

Definition 4 (Practical CJS). $\textsc{cjs}_{pr}(p(\mathbf{X}) \mid\mid q(\mathbf{X}))$ *is*

$$\textsc{cjs}\left(p(X_{\sigma(1)}, \ldots, X_{\sigma(m)}) \mid\mid q(X_{\sigma(1)}, \ldots, X_{\sigma(m)})\right)$$

where $\sigma \in \mathcal{F}$ is a permutation such that $\textsc{cjs}(X_{\sigma(1)}) \geq \ldots \geq \textsc{cjs}(X_{\sigma(m)})$.

In other words, $\textsc{cjs}_{pr}$ chooses the permutation corresponding to the sorting of dimensions in descending order of CJS values. The intuition behind this choice is that the difference between $p(X_i \mid \ldots)$ and $q(X_i \mid \ldots)$ is likely reflected through the difference between $p(X_i)$ and $q(X_i)$. Thus, by ordering dimensions in terms of their CJS values, we can approximate the optimal permutation. Although a greedy heuristic, our experiments reveal that $\textsc{cjs}_{pr}$ works well in practice. For exposition, from now on we simply assume that σ is the *identity mapping function*, i.e. the permutation of dimensions is $X_1, \ldots, X_m$. Following the proof of Theorem 3, we also have that $\textsc{cjs}_{pr}$ is a divergence measure.

Theorem 4. $\textsc{cjs}_{pr}(p(\mathbf{X}) \mid\mid q(\mathbf{X})) \geq 0$ *with equality iff* $p(\mathbf{X}) = q(\mathbf{X})$.

In the remainder of the paper we will consider $\textsc{cjs}_{pr}$ and for readability simply refer to it as CJS.

2.3 Computing CJS

To compute $\textsc{cjs}(p(\mathbf{X}) \mid\mid q(\mathbf{X}))$, we need to compute unconditional and conditional CJS. For the former, suppose that we want to compute $\textsc{cjs}(p(X) \mid\mid q(X))$ for $X \in \mathbf{X}$. Let $v \leq X[1] \leq \ldots \leq X[n_p] \leq V$ be realisations of X drawn from $p(X)$. Further, let $P_{n_p}(x) = \frac{1}{n_p} \sum_{j=1}^{n_p} I(X[j] \leq x)$. Following [15], we have

$$\int P(x)dx = \sum_{j=1}^{n_p - 1} (X[j+1] - X[j]) \frac{j}{n_p} + (V - X[n_p]) \quad .$$

The other terms required for calculating $\textsc{cjs}(p(X) \mid\mid q(X))$ (cf., Definition 1), e.g. $\int Q(x)dx$, are similarly computed. More details can be found in [15].

Computing conditional CJS terms, however, requires pdfs – which are unknown. We resolve this in a non-parametric way using *optimal* discretisation. That is, we first compute $\textsc{cjs}(p(X_1) \mid\mid q(X_1))$. Next, we calculate $\textsc{cjs}(p(X_2 \mid X_1) \mid\mid q(X_2 \mid X_1))$ by searching for the discretisation of X_1 that maximises this term. At step $k \geq 3$, we compute $\textsc{cjs}(p(X_k \mid X_1, \ldots, X_{k-1}) \mid\mid q(X_k \mid X_1, \ldots, X_{k-1}))$ by searching for the discretisation of X_{k-1} that maximises this term. Thus, we only discretise the dimension picked in the previous step and do not re-discretise any earlier chosen dimensions. First and foremost, this increases the efficiency of our algorithm. Second, and more importantly, it facilitates interpretability as we only have to consider one discretisation per dimension.

Next, we show that the discretisation at a step can be done efficiently and optimally by dynamic programming. For simplicity, let $\mathbf{X}' \subset \mathbf{X}$ be the set of dimensions already picked *and* discretised. We denote X as the dimension selected in the previous step but not yet discretised. Let X_c be the dimension selected in this step. Our goal is to find the discretisation of X maximising $\textsc{cjs}(p(X_c \mid \mathbf{X}', X) \mid\mid q(X_c \mid \mathbf{X}', X))$.

To accomplish this, let $X[1] \leq \ldots \leq X[n_p]$ be realisations of X drawn from *the samples of* $p(X)$. We write $X[j, u]$ for $\{X[j], X[j+1], \ldots, X[u]\}$ where $j \leq u$. Note that $X[1, n_p]$ is in fact X. We use

$$\textsc{cjs}(p(X_c \mid \mathbf{X}', \langle X[j, u] \rangle) \mid\mid q(X_c \mid \mathbf{X}', \langle X[j, u] \rangle))$$

to denote $\textsc{cjs}(p(X_c \mid \mathbf{X}') \mid\mid q(X_c \mid \mathbf{X}'))$ computed using the $(u - j + 1)$ samples of $p(X)$ corresponding to $X[j]$ to $X[u]$, projected onto X. For $1 \leq l \leq u \leq n_p$, we write

$$f(u, l) = \max_{dsc : |dsc| = l} \textsc{cjs}\left(p(X_c \mid \mathbf{X}', X^{dsc}[1, u]) \mid\mid q(X_c \mid \mathbf{X}', X^{dsc}[1, u])\right)$$

where dsc is a discretisation of $X[1, u]$, $|dsc|$ is its number of bins, and $X^{dsc}[1, u]$ is the discretised version of $X[1, u]$ produced by dsc. For $1 < l \leq u \leq n_p$, we have

Theorem 5. *$f(u, l) = \max_{j \in [l-1, u)} \mathcal{A}_j$ where*

$$\mathcal{A}_j = \frac{j}{u} f(j, l-1) + \frac{u-j}{u} \textsc{cjs}\left(p(X_c \mid \mathbf{X}', \langle X[j+1, u] \rangle) \mid\mid q(X_c \mid \mathbf{X}', \langle X[j+1, u] \rangle)\right)$$

Proof. We postpone the proof to Appendix A. □

Theorem 5 shows that the optimal discretisation of $X[1, u]$ can be derived from that of $X[1, j]$ with $j < u$. This allows us to design a dynamic programming algorithm to find the discretisation of X maximising $\text{CJS}(p(X_c \mid \mathbf{X}', X) \mid\mid q(X_c \mid \mathbf{X}', X))$.

2.4 Complexity Analysis

We now discuss the time complexity of computing $\text{CJS}(p(\mathbf{X}) \mid\mid q(\mathbf{X}))$. When discretising a dimension $X \in \mathbf{X}$, if we use its original set of data samples as cut points, the time complexity of solving dynamic programming is $O(n_p^2)$, rather restrictive for large data. Most cut points, however, will not be used in the optimal discretisation. To gain efficiency, we can hence impose a maximum grid size $max\_grid = n_p^\epsilon$ and limit the number of cut points to $c \times max\_grid$ with $c > 1$. To find these candidate cut points, we follow Reshef et al. [20] and apply equal-frequency binning on X with the number of bins equal to $(c \times max\_grid + 1)$. Note that this pre-processing trades off accuracy for efficiency. Other types of pre-processing are left for future work.

Regarding ϵ and c, the larger they are, the more candidate discretisations we consider, and hence, at a higher the computational cost, the better the result. Our empirical results show that $\epsilon = 0.5$ and $c = 2$ offers a good balance between quality and efficiency, and we will use these values in the experiments. The cost of discretising each dimension X then is $O(n_p)$. The overall complexity of computing $\text{CJS}(p(\mathbf{X}) \mid\mid q(\mathbf{X}))$ is therefore $O(m \times n_p)$. Similarly, the complexity of computing $\text{CJS}(q(\mathbf{X}) \mid\mid p(\mathbf{X}))$ is $O(m \times n_q)$.

2.5 Summing Up

We note that CJS is asymmetric. To have a symmetric distance, we use

$$\text{CJS}_{sym}(p(\mathbf{X}) \mid\mid q(\mathbf{X})) = \text{CJS}(p(\mathbf{X}) \mid\mid q(\mathbf{X})) + \text{CJS}(q(\mathbf{X}) \mid\mid p(\mathbf{X})) \quad .$$

In addition, we present two important properties pertaining specifically to univariate CJS_{sym}. Although in the interest of space we will not explore these properties empirically, but they may be important to know for other applications of our measure.

Theorem 6. $\text{CJS}_{sym}(p(X) \mid\mid q(X)) \leq \int (P(x) + Q(x))\, dx$.

Proof. We postpone the proof to Appendix A. □

Theorem 7. *Univariate* $\sqrt{\text{CJS}_{sym}}$ *is a metric.*

Proof. We follow the proof of Theorem 1 in [7]. □

Theorem 6 tells us that the value of univariate CJS_{sym} is bounded above, which facilitates interpretation [11]. Theorem 7 on the other hand says that the square root of univariate CJS_{sym} is a metric distance. This is beneficial for, e.g. query optimisation in multimedia databases.

3 Related Work

Many divergence measures have been proposed in the literature. Besides Kullback-Leibler and Jensen-Shannon, other well-known divergence measures include the Kolmogorov-Smirnov test (KS), the Cramér-von Mises criterion (CM), Earth Mover's Distance (EMD), and the quadratic measure of divergence (QR) [13]. Each has its own strengths and weaknesses – most particularly w.r.t. exploratory analysis. For example, multivariate KS, CM, EMD, and QR all operate on the joint distributions over *all* dimensions. Thus, they inherently suffer from the curse of dimensionality, which reduces their statistical power when applied on non-trivial numbers of dimensions. In addition, EMD needs probability mass functions (pmfs). While readily available for discrete data, real-valued data first needs to be discretised. There currently exists no discretisation method that directly optimises EMD, however, by which the results may turn out ad hoc. Recently, Perez-Cruz [17] studied how to estimate KL using cdfs. Park et al. [16] proposed CKL, redefining KL by replacing pdfs with cdfs. While computable on empirical data, as for regular KL it may be undefined when $p(\mathbf{x}) = 0$ or $q(\mathbf{x}) = 0$ for some $\mathbf{x} \in dom(\mathbf{X})$. Further, CKL was originally proposed as a univariate measure. Wang et al. [24] are the first to formulate JS using cdfs. However, their CJS relies on joint cdfs and hence suffers from the curse of dimensionality.

Many data mining tasks require divergence measures. For instance, for change detection on time-series, it is necessary to test whether two windows of data are sampled from the same underlying distribution. Song et al. [23] proposed such a test, using Gaussian kernels to approximate the data distribution – including the joint distribution over all dimensions. Generalisations of KL computed using Gaussian kernels have shown to be powerful alternatives [8, 12]. KL is also used for anomaly detection in time series, where we can compute an anomaly score for a window against the reference data set [18]. In interaction-preserving discretisation we need to assess how different (multivariate) distributions are between two consecutive bins. This can be done through contrast mining [3], or by using QR [13]. In multi-target subgroup discovery, also known as exceptional model mining [10], we need to compare the distributions of subgroup against that of its complement data set. Leman et al. use a quadratic measure of divergence [10], whereas Duivesteijn et al. consider the edit distance between Bayesian networks [6]. In Section 4, we will consider the efficacy of CJS for each of these areas.

Nguyen et al. [15] proposed a correlation measure inspired by factorised KL using cumulative entropy [19]. Although it permits reliable non-parametric computation on empirical data, it uses ad hoc clustering to compute conditional entropies. Nguyen et al. [14] showed that these are inferior to optimal discretisation, in their case for total correlation. In CJS we use the same general idea of optimal discretisation, yet the specifics for measuring divergence are nontrivial and had to be developed from scratch.

4 Experiments

Next, we empirically evaluate CJS. In particular, we will evaluate the statistical power at which it quantifies differences between data distributions, and its scalability to data size and dimensionality. In addition, we evaluate its performance in four exploratory data mining tasks. We implemented CJS in Java, and make our code available for research purposes.[1] All experiments were performed single-threaded on an Intel(R) Core(TM) i7-4600U CPU with 16GB RAM. We report wall-clock running times.

We compare CJS to MG [23] and RSIF [12], two measures of distribution difference recently proposed for change detection on time series. In short, to compare two samples S_p and S_q, MG randomly splits S_p into S_p^1 and S_p^2. Next, it uses S_p^1 to model the distribution of data. Then it fits S_p^2 and S_q into the model. The difference in their fitness scores is regarded as the difference between S_p and S_q. RSIF on the other hand uses a non-factorised variant of KL divergence. To compute this divergence, it estimates the ratio $\frac{p(\mathbf{X})}{q(\mathbf{X})}$. As third baseline, we consider QR, a quadratic measure of distribution difference recently proposed by Nguyen et al. [13]. It works on $P(\mathbf{X})$ and $Q(\mathbf{X})$, i.e. the cdfs of all dimensions. Note that by their definition these three competitors are prone to the curse of dimensionality. Finally, we include CKL, extended to the multivariate setting similarly to CJS.

4.1 Statistical Power

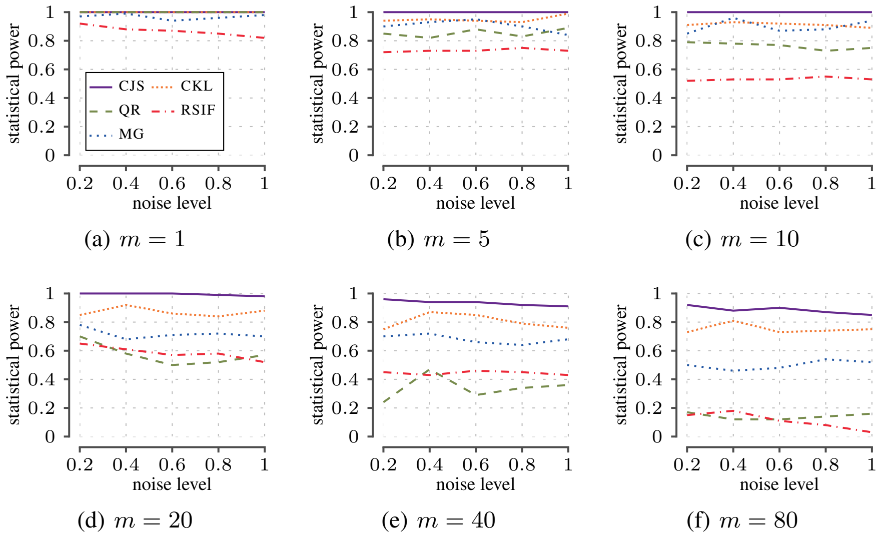
Our aim here is to examine if our measure is really suitable for quantifying the difference between two data distributions. For this purpose, we perform statistical tests using synthetic data. To this end, the null hypothesis is that the two distributions are similar. To determine the cutoff for testing the null hypothesis, we first generate 100 pairs of data sets of the same size (n) and dimensionality (m), and having the same distribution f_1. Next, we compute the divergence score for each pair. Subsequently, we set the cutoff according to the significance level $\alpha = 0.05$. We then generate 100 pairs of data sets, again with the same n and m. However, two data sets in such a pair have different distributions. One follows distribution f_1 while the other follows distribution f_2. The power of the measure is the proportion of the 100 new pairs of data sets whose divergence scores exceed the cutoff. We simulate a noisy setting by adding Gaussian noise to the data. We show the results in Fig. 1 for $n = 1\,000$ and varying over m with f_1 and f_2 two Gaussian distributions with different mean vectors and covariance matrices. For other data sizes and distributions we observe the same trend.

Inspecting these results, we find that CJS obtains higher statistical power than other measures. Moreover, it is very stable across dimensionality and noise. Other measures, especially QR and RSIF, deteriorate with high dimensionality. Overall, we find that CJS reliably measures the divergence of distributions, regardless of dimensionality or noise.

4.2 Scalability

Next, we study the scalability of our measures with respect to the data size n and dimensionality m. For scalability to n, we generate data sets with $m = 10$ and n varied from $1\,000$ to $20\,000$. For scalability to m, we generate data sets with $n = 1\,000$ and m varied from 1 to 80. We present the results in Fig. 2. We observe that our measure

[1] `http://eda.mmci.uni-saarland.de/cjs/`



(a) $m = 1$ (b) $m = 5$ (c) $m = 10$

(d) $m = 20$ (e) $m = 40$ (f) $m = 80$

Fig. 1. [Higher is better] Statistical power vs. dimensionality of CJS, CKL, QR, RSIF, and MG on synthetic data sets. Overall, CJS achieves the best statistical power across different dimensionality and noise levels.

is efficient. It scales similarly as CKL, QR, and MG, and much better than RSIF. Combining this with our results regarding statistical power, we conclude that CJS yields the best balance between quality and efficiency.

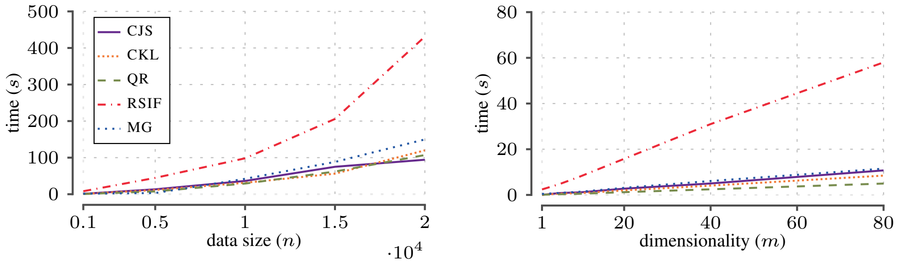
The results show that CJS outperforms CKL while the two have similar runtime. We therefore exclude CKL in the remainder.

4.3 Change Detection on Time Series

Divergence measures are widely used for change detection on time series [4, 12, 21]. The main idea is that given a window size W, at each time instant t, we measure the difference between the distribution of data over the interval $[t - W, t)$ to that over the interval $[t, t + W)$. A large difference is an indicator that a change may have occurred. The quality of change detection is thus dependent on the quality of the measure. In this experiment, we apply CJS in change detection. In particular, we use it in the retrospective change detection model proposed by Liu et al. [12].

As data, we use the PAMAP data set,[2] which contains human activity monitoring data. Essentially, it consists of data recorded from sensors attached to 9 human subjects. Each subject performs different types of activities, e.g. *standing*, *walking*, *running*, and each activity is represented by 51 sensor readings recorded per second. Since each subject has different physical characteristics, we consider his/her data to be a separate data set. One data set is very small, so we discard it. We hence consider 8 time series over

[2] http://www.pamap.org/demo.html



(a) Runtime against data size

(b) Runtime against dimensionality

Fig. 2. [Lower is better] Runtime scalability of CJS, CKL, QR, RSIF, and MG on synthetic data sets. Overall, CJS scales similarly to CKL, QR, and MG and much better than RSIF.

Data	**CJS**	**QR**	**RSIF**	**MG**
Subject 1	**0.972**	0.658	0.662	0.775
Subject 2	**0.977**	0.669	0.694	0.782
Subject 3	**0.971**	0.663	0.857	0.954
Subject 4	**0.973**	0.641	0.662	0.642
Subject 5	**0.988**	0.678	0.756	0.850
Subject 6	**0.977**	0.662	0.497	0.550
Subject 7	**0.978**	0.646	0.782	0.705
Subject 8	**0.973**	0.741	0.552	0.424
Average	**0.976**	0.670	0.683	0.710

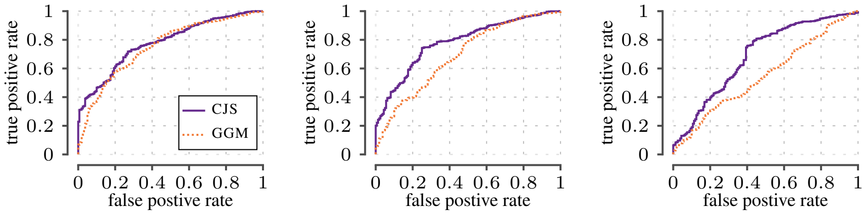
Table 1. [Higher is better] AUC scores of CJS, QR, RSIF, and MG in time-series change detection on PAMAP data sets. Highest values are in **bold**. Overall, CJS yields the best accuracy across all subjects.

51 dimensions with in the order of $100\,000$ time points. In each time series, the time instants when the respective subject changes his/her activities are regarded as change points. As the change points are known, we evaluate how well each measure tested retrieves these cut points. It is expected that each measure should assign higher difference scores at the change points in comparison to other normal time instants. As performance metric we construct Receiver Operating Characteristic (ROC) curves and consider the Area Under the ROC curve (AUC) [8, 12, 23].

Table 1 gives the results. We see that CJS consistently achieves the best AUC over all subjects. Moreover, it outperforms its competitors with relatively large margins.

4.4 Anomaly Detection on Time Series

Closely related to change detection is anomaly detection [2, 18]. The core idea is that a reference data set is available as training data. For example, obtained for instance from historical records. It is used for building a statistical model capturing the generation process of normal data. Then, a window is slid along the test time series to compute



(a) training size = 200 (b) training size = 300 (c) training size = 400

Fig. 3. [Higher is better] ROC curves of CJS and GGM regarding time-series anomaly detection on the TEP data set. AUC scores of CJS are respectively 0.780, 0.783, and 0.689. AUC scores of GGM are respectively 0.753, 0.691, and 0.552. Overall, CJS outperforms GGM.

the anomaly score for each time instant, using the model constructed. With CJS, we can perform the same task by simply comparing the distribution over a window against that of the reference set. That is, no model construction is required. In contrast to GGM [18] – a state of the art method for anomaly detection in time series – CJS can be considered as a 'lazy' detector. We will assess how CJS performs against GGM. For this, we use the TEP data set, as it was used by Qiu et al. [18]. It contains information on an industrial production process. The data has 52 dimensions. Following their setup, we set the window size to 10. We vary the size of the training set to assess stability.

Fig. 3 presents the results. We see that CJS outperfoms GGM at its own game. In particular, we see that CJS is less sensitive to the size of the training set than GGM, which could be attributed to its 'lazy' approach. Overall, the conclusion is that CJS reliably measures the difference of multivariate distributions.

4.5 Multivariate Discretisation

When discretising multivariate data the key goal is to discretise the data such that the output data preserves the most important multivariate interactions in the input data [3, 13]. Only when we do so it will be possible to use techniques that require discrete data – such as pattern mining – to pick up on truly interesting correlations. One of the major components of interaction-preserving discretisation is to measure the difference of data distributions in different bins. The difference scores are then used to decide if bin merge takes place or not.

In principle, the better such measure, the better correlations can be maintained. For example, the better pattern-based compressors such as COMPREX [1] can compress it. In this experiment, we apply CJS in IPD [13] – a state of the art technique for interaction-preserving discretisation. To evaluate, we apply COMPREX to the discretised data and compare the total encoded size. We compare against original IPD, which uses QR. For testing purposes, we use 6 public data sets available in the UCI Repository.

We display the results in Fig. 4. The plot shows the relative compression rates with IPD as the bases, per data set. Please note that lower compression costs are better. Going over the results, we can see that CJS improves the performance IPD in 4 out of 6 data

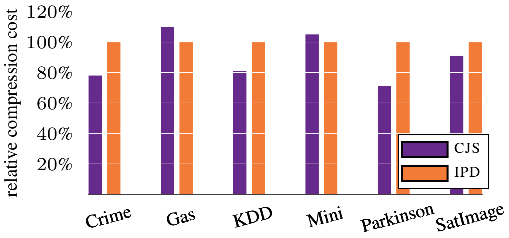


Fig. 4. [Lower is better] Relative compression costs of CJS and IPD in interaction-preserving discretisation. COMPREX [1] is the compressor. The compression costs of IPD are the bases. Overall, CJS outperforms IPD.

sets. This implies that CJS reliably assesses the difference of multivariate distributions in different bins [13].

4.6 Multi-Target Subgroup Discovery

In subgroup discovery we are after finding queries – patterns – that identify subgroups of data points for which the distribution of some target attribute varies strongly compare to either the complement, or the whole data. As the name implies, in multi-target subgroup discovery we do not consider a univariate targets, but multivariate ones.

Formally, let us consider a data set $\mathbf{D}$ with attributes $A_1, \ldots, A_k$ and targets $T_1, \ldots, T_l$. A subgroup $\mathcal{S}$ on $\mathbf{D}$ is characterized by condition(s) imposed on some attribute(s). A condition on an attribute A has the form of an interval. The subset of $\mathbf{D}$ corresponding to $\mathcal{S}$ is denoted as $\mathbf{D}_\mathcal{S}$. The set of remaining data points, the complement set, is $\overline{\mathbf{D}}_\mathcal{S} = \mathbf{D} \setminus \mathbf{D}_\mathcal{S}$. Within subgroup discovery, exceptional model mining is concerned with detecting $\mathcal{S}$ such that $p(T_1, \ldots, T_l \mid \mathbf{D}_\mathcal{S})$ is different from $p(T_1, \ldots, T_l \mid \overline{\mathbf{D}}_\mathcal{S})$ [6, 10]. The higher the difference, the better.

In this experiment, we use CJS for quantifying the distribution divergence non-parametrically. Apart from that, we apply as-is the search algorithm proposed in [6] for discovering high quality subgroups. As data sets, we use 3 public ones. Two from the UCI Repository, namely, the *Bike* dataset of 731 data points over 6 attributes with 2 targets, and the *Energy* dataset of 768 rows over 8 attributes also with 2 targets. Third, we consider the *Chemnitz* dataset of 1440 rows over 3 attributes and with 7 targets.[3] Our objective here is to see if CJS can assist in discovering interesting subgroups on these data sets. The representative subgroups on three data sets are in Table 2 (all subgroups are significant at significance level $\alpha = 0.05$).

Going over the results, we see CJS to detect subgroups having different distribution in targets compared to that of their respective complement set. For instance, on *Bike* we discover the subgroup *temperature* $\geq 6.5 \wedge$ *temperature* < 10.7. In this subgroup, we find that its numbers of registered and non-registered bikers are significantly lower than those of its complement set. This is intuitively understandable, as at these low temperatures one expects to see only a few bikers, and especially few casual ones. In

[3] http://www.mathe.tu-freiberg.de/Stoyan/umwdat.html

contrast, for the subgroup *temperature* $\geq 27.1 \wedge$ *temperature* < 31.2, the numbers of bikers in both targets are very high. This again is intuitively understandable.

From the *Energy* data, we find that the two subgroups *surface area* $\geq 624.8 \wedge$ *surface area* < 661.5 and *roof area* < 124.0 have much higher heating and cooling loads compared to their complement sets.

The previous two data sets contain 2 targets only. In contrast, Chemnitz data set has 7 targets, which poses a more challenging task. Nevertheless, with CJS we can detect informative subgroups as it can capture divergences between distributions that are involved in different numbers of targets – not all target attributes have to be 'divergent' at the same time, after all. In particular, the subgroup *temperature* $\geq 4.25 \wedge$ *temperature* < 7.5 has its divergence traced back to five targets. On the other hand, there are only two targets responsible for the divergence of the subgroup *wind* < -0.75.

Overall, we find that CJS can be successfully applied to non-parametrically discover subgroups in real-world data with multiple targets.

5 Discussion

The experiments show that CJS is efficient and obtains high statistical power in detecting divergence for varying dimensionality and noise levels. Further, we demonstrated that CJS is well-suited for a wide range of exploratory tasks, namely time-series change detection and anomaly detection, interaction-preserving discretisation, and multi-target subgroup discovery. The improvement in performance of CJS over existing measures can be traced back to its three main properties: (a) it does not make any assumption on the relation between two distributions, (b) it allows non-parametric computation on empirical data, and (c) it is less sensitive to the curse of dimensionality.

Yet, there is room for alternative methods as well as further improvements. For instance, in this paper, we pursue the non-parametric setting. As long as the knowledge on data distributions is known, one can resort to parametric methods to compute other divergence measures, e.g. KL and JS. A promising direction is to extend CJS to heterogeneous data types. That is, in addition to numerical data, we can consider categorical data as well. A possible solution to this end is to combine JS and CJS. More in particular, JS is used to handle categorical data; CJS is used for numerical data; and discretisation can be used to bridge both worlds. The details, however, are beyond the scope of this work. As future work, we also plan to develop new subgroup discovery methods that integrate CJS more deeply into the mining process. This will help us to better exploit the capability of CJS in this interesting branch of exploratory analysis.

6 Conclusion

In this paper, we proposed CJS, an information-theoretic divergence measure to quantify the difference of two distributions. In short, CJS requires neither assumptions on the forms of distributions nor their relation. Further, it permits efficient non-parametric computation on empirical data. Extensive experiments on both synthetic and real-world data showed that our measure outperforms the state of the art in both statistical power and efficiency in a wide range of exploratory tasks.

		Mean	
Data	**Target**	**subgroup** ($\mathbf{D}_{\mathcal{S}}$)	**complement** ($\overline{\mathbf{D}}_{\mathcal{S}}$)
Bike	$6.5 \leq$ *temperature* < 10.7 (support = 63)		
	registered bikers	166	913
	non-registered bikers	1 889	3 840
	$27.1 \leq$ *temperature* < 31.2 (support = 127)		
	registered bikers	1 347	743
	non-registered bikers	4 406	3 499
Energy	$624.8 \leq$ *surface area* < 661.5 (support = 128)		
	heating	38.6	20.8
	cooling	40.2	23.1
	roof area < 124.0 (support = 192)		
	heating	31.6	19.1
	cooling	33.1	21.7
Chemnitz	$4.25 \leq$ *temperature* < 7.5 (support = 370)		
	dust	53.5	109.7
	SO_2	80.6	184.4
	NO_2	20.4	41.4
	NO_x	50.2	94.3
	wind < -0.75 (support = 395)		
	NO	69.0	39.0
	NO_x	106.4	74.1

Table 2. Representative subgroups discovered by CJS on Bike, Energy, and Chemnitz data sets. On Chemnitz, only targets where the divergence is large are shown. Overall, CJS helps detect high quality and informative subgroups on all three data sets.

Acknowledgements

The authors thank the anonymous reviewers for insightful comments. Hoang-Vu Nguyen and Jilles Vreeken are supported by the Cluster of Excellence “Multimodal Computing and Interaction” within the Excellence Initiative of the German Federal Government.

References

1. L. Akoglu, H. Tong, J. Vreeken, and C. Faloutsos. COMPREX: Compression based anomaly detection. In *CIKM*. ACM, 2012.
2. A. Arnold, Y. Liu, and N. Abe. Temporal causal modeling with graphical granger methods. In *KDD*, pages 66–75, 2007.

3. S. D. Bay. Multivariate discretization for set mining. *Knowledge and Information Systems*, 3(4):491–512, 2001.
4. V. Chandola and R. R. Vatsavai. A gaussian process based online change detection algorithm for monitoring periodic time series. In *SDM*, pages 95–106, 2011.
5. T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience New York, 2006.
6. W. Duivesteijn, A. J. Knobbe, A. Feelders, and M. van Leeuwen. Subgroup discovery meets bayesian networks – an exceptional model mining approach. In *ICDM*, pages 158–167, 2010.
7. D. M. Endres and J. E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, 2003.
8. Y. Kawahara and M. Sugiyama. Change-point detection in time-series data by direct density-ratio estimation. In *SDM*, pages 389–400, 2009.
9. S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.
10. D. Leman, A. Feelders, and A. J. Knobbe. Exceptional model mining. In *ECML/PKDD*, pages 1–16, 2008.
11. J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.
12. S. Liu, M. Yamada, N. Collier, and M. Sugiyama. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43:72–83, 2013.
13. H. V. Nguyen, E. Müller, J. Vreeken, and K. Böhm. Unsupervised interaction-preserving discretization of multivariate data. *Data Min. Knowl. Discov.*, 28(5-6):1366–1397, 2014.
14. H. V. Nguyen, E. Müller, J. Vreeken, P. Efros, and K. Böhm. Multivariate maximal correlation analysis. In *ICML*, pages 775–783, 2014.
15. H. V. Nguyen, E. Müller, J. Vreeken, F. Keller, and K. Böhm. CMI: An information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In *SDM*, pages 198–206, 2013.
16. S. Park, M. Rao, and D. W. Shin. On cumulative residual Kullback-Leibler information. *Statistics and Probability Letters*, 82:2025–2032, 2012.
17. F. Perez-Cruz. Kullback-leibler divergence estimation of continuous distributions. In *ISIT*, pages 1666–1670. IEEE, 2008.
18. H. Qiu, Y. Liu, N. A. Subrahmanya, and W. Li. Granger causality for time-series anomaly detection. In *ICDM*, pages 1074–1079, 2012.
19. M. Rao, Y. Chen, B. C. Vemuri, and F. Wang. Cumulative residual entropy: A new measure of information. *IEEE Transactions on Information Theory*, 50(6):1220–1228, 2004.
20. D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti. Detecting novel associations in large data sets. *Science*, 334(6062):1518–1524, 2011.
21. Y. Saatci, R. D. Turner, and C. E. Rasmussen. Gaussian process change point models. In *ICML*, pages 927–934, 2010.
22. D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons Inc, New York, 1992.
23. X. Song, M. Wu, C. M. Jermaine, and S. Ranka. Statistical change detection for multi-dimensional data. In *KDD*, pages 667–676, 2007.
24. F. Wang, B. C. Vemuri, and A. Rangarajan. Groupwise point pattern registration using a novel cdf-based jensen-shannon divergence. In *CVPR*, pages 1283–1288, 2006.

A Appendix

Proof (Theorem 3). First, we prove that

$$\textsc{cjs}(p(X_1,\ldots,X_m) \mid\mid q(X_1,\ldots,X_m)) = 0$$

iff $p(\mathbf{X}) = q(\mathbf{X})$.

($\Rightarrow$): From Theorem 1, we have $p(X_i \mid X_1,\ldots,X_{i-1}) = q(X_i \mid X_1,\ldots,X_{i-1})$ for $i \in [1,m]$. This implies $p(\mathbf{X}) = q(\mathbf{X})$.

($\Leftarrow$): When $p(\mathbf{X}) = q(\mathbf{X})$, we have $p(X_i \mid X_1,\ldots,X_{i-1}) = q(X_i \mid X_1,\ldots,X_{i-1})$ for $i \in [1,m]$. Thus, following Theorem 1, we have that

$$\textsc{cjs}(p(X_1,\ldots,X_m) \mid\mid q(X_1,\ldots,X_m)) = 0 \quad .$$

The above proof in fact holds for *any* permutation. Thus, $\textsc{cjs}(p(\mathbf{X}) \mid\mid q(\mathbf{X})) \geq 0$ with equality iff $p(\mathbf{X}) = q(\mathbf{X})$. □

Proof (Theorem 5). Let

$$dsc^* = \arg\max_{dsc:|dsc|=l} \textsc{cjs}\left(p(X_c \mid \mathbf{X}', X^{dsc}[1,u]) \mid\mid q(X_c \mid \mathbf{X}', X^{dsc}[1,u])\right) .$$

We denote l bins that dsc^* generates on X as $b(X)_1,\ldots,b(X)_l$. We write $|b(X)_t|$ as the number of values of X in $b(X)_t$. For each $X'_i \in \mathbf{X}'$, we denote its bins as $b(X'_i)_1,\ldots,b(X'_i)_{n'_i}$.

Further, let $c_z = \sum\limits_{i=1}^{z} |b(X)_t|$. Note that each bin of X is non-empty, i.e. $c_z \geq z$. We use $\textsc{cjs}\,(p(X_c \mid \mathbf{X}', b_t) \mid\mid q(X_c \mid \mathbf{X}', b_t))$ to denote $\textsc{cjs}\,(p(X_c \mid \mathbf{X}') \mid\mid q(X_c \mid \mathbf{X}'))$ computed using the samples of $p(\mathbf{X})$ corresponding to the realizations of X in $b(X)_t$, projected onto Y and $\mathbf{X}'$. We write $|(t,t_1,\ldots,t_k)|$ as the number of points in the cell made up by bins $b(X)_t, b(X'_1)_{t_1},\ldots,b(X'_{|\mathbf{X}'|})_{t_{|\mathbf{X}'|}}$.

We have: $f(u, l)$

$$
\begin{aligned}
&= \sum_{t=1}^{l} \sum_{t_1=1}^{n'_1} \cdots \sum_{t_{|\mathbf{X}'|}=1}^{n'_{|\mathbf{X}'|}} \frac{|(t, t_1, \ldots, t_k)|}{u} \times \\
&\qquad \textsc{cjs}\left(p(X_c \mid b(E)_t, b(X'_1)_{t_1}, \ldots, b(X'_{|\mathbf{X}'|})_{t_{|\mathbf{X}'|}}) \mid\mid q(X_c \mid b(E)_t, b(X'_1)_{t_1}, \ldots, b(X'_{|\mathbf{X}'|})_{t_{|\mathbf{X}'|}})\right) \\
&= \sum_{t=1}^{l-1} \sum_{t_1=1}^{n'_1} \cdots \sum_{t_{|\mathbf{X}'|}=1}^{n'_{|\mathbf{X}'|}} \frac{|(t, t_1, \ldots, t_k)|}{u} \times \\
&\qquad \textsc{cjs}\left(p(X_c \mid b(E)_t, b(X'_1)_{t_1}, \ldots, b(X'_{|\mathbf{X}'|})_{t_{|\mathbf{X}'|}}) \mid\mid q(X_c \mid b(E)_t, b(X'_1)_{t_1}, \ldots, b(X'_{|\mathbf{X}'|})_{t_{|\mathbf{X}'|}})\right) \\
&\qquad + \frac{|b_l|}{u} \textsc{cjs}\left(p(X_c \mid \mathbf{X}', b_l) \mid\mid q(X_c \mid \mathbf{X}', b_l)\right) \\
&= \frac{c_{l-1}}{u} \sum_{t=1}^{l-1} \sum_{t_1=1}^{n'_1} \cdots \sum_{t_{|\mathbf{X}'|}=1}^{n'_{|\mathbf{X}'|}} \frac{|(t, t_1, \ldots, t_k)|}{u} \times \\
&\qquad \textsc{cjs}\left(p(X_c \mid b(E)_t, b(X'_1)_{t_1}, \ldots, b(X'_{|\mathbf{X}'|})_{t_{|\mathbf{X}'|}}) \mid\mid q(X_c \mid b(E)_t, b(X'_1)_{t_1}, \ldots, b(X'_{|\mathbf{X}'|})_{t_{|\mathbf{X}'|}})\right) \\
&\qquad + \frac{u - c_{l-1}}{u} \textsc{cjs}\left(p(X_c \mid \mathbf{X}', \langle X[c_{l-1}+1, u]\rangle) \mid\mid q(X_c \mid \mathbf{X}', \langle X[c_{l-1}+1, u]\rangle)\right) \\
&= \frac{c_{l-1}}{u} f(c_{l-1}, l-1) \\
&\qquad + \frac{u - c_{l-1}}{u} \textsc{cjs}\left(p(X_c \mid \mathbf{X}', \langle X[c_{l-1}+1, u]\rangle) \mid\mid q(X_c \mid \mathbf{X}', \langle X[c_{l-1}+1, u]\rangle)\right).
\end{aligned}
$$

In the last line,

$$
\begin{aligned}
&\sum_{t=1}^{l-1} \sum_{t_1=1}^{n'_1} \cdots \sum_{t_{|\mathbf{X}'|}=1}^{n'_{|\mathbf{X}'|}} \frac{|(t, t_1, \ldots, t_k)|}{u} \times \\
&\qquad \textsc{cjs}\left(p(X_c \mid b(E)_t, b(X'_1)_{t_1}, \ldots, b(X'_{|\mathbf{X}'|})_{t_{|\mathbf{X}'|}}) \mid\mid q(X_c \mid b(E)_t, b(X'_1)_{t_1}, \ldots, b(X'_{|\mathbf{X}'|})_{t_{|\mathbf{X}'|}}\right)
\end{aligned}
$$

is equal to $f(c_{l-1}, l-1)$ because otherwise, we could increase $f(u, l)$ by choosing a different discretization of $X[1, c_{l-1}]$ into $l - 1$ bins. This in turn contradicts our definition of $f(u, l)$. Since $c_{l-1} \in [l-1, u)$ and $f(u, l)$ is maximal over all $j \in [l-1, u)$, we arrive at the final result. □

Proof (Theorem 6). We have:

$$
\begin{aligned}
&\textsc{cjs}_{sym}(p(X) \mid\mid q(X)) \\
&= \int P(x) \log \frac{P(x)}{\frac{1}{2}P(x) + \frac{1}{2}Q(x)} dx + \int Q(x) \log \frac{P(x)}{\frac{1}{2}P(x) + \frac{1}{2}Q(x)} dx \\
&= \int (P(x) + Q(x)) \left(\frac{P(x)}{P(x)+Q(x)} \log \frac{2P(x)}{P(x)+Q(x)} + \frac{Q(x)}{P(x)+Q(x)} \log \frac{2Q(x)}{P(x)+Q(x)} \right) dx \\
&= \int (P(x) + Q(x)) \left(1 - H\left(\frac{P(x)}{P(x)+Q(x)}, \frac{Q(x)}{P(x)+Q(x)} \right) \right) dx.
\end{aligned}
$$

As $H(.) \geq 0$, we arrive at: $\textsc{cjs}_{sym}(p(X) \mid\mid q(X)) \leq \int (P(x) + Q(x))\, dx$. $\square$