The Kernel Density Quantile Transformation

Anonymous authors
Paper under double-blind review

Abstract

Feature preprocessing continues to play a critical role when applying machine learning and statistical methods to tabular data. In this paper, we propose the use of the kernel-density quantile transformation as a feature preprocessing step. We describe and demonstrate how the kernel-density quantile transformation can be used as a simple drop-in replacement for either min-max scaling or quantile transformation, combining the advantages of each of those preprocessing methods. Furthermore, we show that, in addition to its utility in supervised machine learning, the kernel-density transformation can be profitably applied to statistical data analysis, particularly in correlation analysis and univariate clustering.

1 Introduction

Feature preprocessing is a ubiquitous workhorse in applied machine learning and statistics, particularly for structured (tabular) data. Two of the most common preprocessing methods are min-max scaling, which linearly rescales each feature to have the range $[0, 1]$, and quantile transformation (Bartlett, 1947; Van der Waerden, 1952), which nonlinearly maps each feature to its quantiles, also lying in the range $[0, 1]$. Min-max scaling preserves the shape of each feature's distribution, but is not robust to the effect of outliers, such that output features' variances are not identical. (Other linear scaling methods, such as z-score standardization, can guarantee output uniform variances at the cost of non-identical output ranges.) On the other hand, quantile transformation reduces the effect of outliers, and guarantees identical output variances (and indeed, all moments) as well as ranges; however, all information about the shape of feature distributions is lost.

In this paper, we observe that, by computing definite integrals over the kernel density estimator (KDE) (Rosenblatt, 1956; Parzen, 1962; Silverman, 1986) and tuning the kernel bandwidth, we may construct a tunable "happy medium" between min-max scaling and quantile transformation. We demonstrate that the kernel density quantile transformation is easily and broadly applicable across a range of problems in machine learning and statistics.

We hasten to point out that kernel density estimators of quantiles have been previously proposed and extensively analyzed (Yamato, 1973; Sheather, 1990; Kulczycki & DaWidowicz, 1999). However, previous works used the KDE to yield quantile estimators with superior statistical qualities. In particular, statistical consistency is desired for such estimators, and so the kernel bandwidth h is chosen such that $h \to 0$ as sample size $N \to \infty$. However, in our case, we are not interested in estimating the true quantiles or the cumulative distribution function (c.d.f.), but instead use the KDE merely to construct a preprocessing transformation for downstream prediction and data analysis tasks. In fact, as we will see later, we choose h to be large and non-vanishing, so that our preprocessed features deviate substantially from the estimated quantiles.

Our contributions are as follows. First, we propose the kernel density quantile transformation as a feature preprocessing method, and provide a computationally-efficient approximation algorithm. Second, we demonstrate the use of kernel-density quantiles in correlation analysis, enabling a useful compromise between Pearson's r and Spearman's ρ. Third, we propose a discretization method for univariate data, based on computing local extrema in the kernel density estimator of the kernel density quantiles.

2 Methods

2.1 Preliminaries

Our approach is inspired by the behavior of min-max scaling and quantile transformation, which we briefly describe below.

The min-max transformation can be derived by considering a random variable X defined over some known range $[U, V]$. In order to transform this variable onto the range $[0, 1]$, one may define the mapping $S : \mathbb{R} \to [0, 1]$ defined as $x \to S(x) := \frac{x-U}{V-U}$, with the upper and lower bounds achieved at $x = U$ and $x = V$, respectively. In practice, one typically observes N random samples $X_1, \ldots, X_N$, which may be sorted into order statistics $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(N)}$. Substituting the minimum and maximum for U and V respectively, we obtain the min-max scaling function

$$\hat{S}_N(x) := \begin{cases} \frac{x-X_{(1)}}{X_{(N)}-X_{(1)}}, & X_{(1)} \leq x \leq X_{(N)} \\ 0, & x \leq X_{(1)} \\ 1, & X_{(N)} \leq x. \end{cases} \tag{1}$$

Quantile transformation can be derived by considering a random variable X with known continuous and strictly monotonically increasing c.d.f. $F_X : \mathbb{R} \to [0, 1]$. The quantile transformation (to be distinguished from the quantile function, or inverse c.d.f.) is identical to the c.d.f, simply mapping each input value x to $F_X(x) := P[X \leq x]$. One typically observes N random samples $X_1, \ldots, X_N$ and obtains an empirical c.d.f. as follows:

$$\hat{F}_N(x) = \frac{1}{N} \sum_{n=1}^{N} I\{X_n \leq x\} := \hat{P}[X \leq x]. \tag{2}$$

In practice, the quantile transformation also requires ensuring sensible behavior despite ties in the observed data.

2.2 The Kernel Density Quantile Transformation

Recall the Gaussian kernel density estimator (KDE) (Silverman, 1986), with the following density:

$$\widehat{f}_h(x) = \frac{1}{N} \sum_{i=n}^{N} K_h(x - X_n), \tag{3}$$

where Gaussian kernel $K(z) = \exp(-z^2/2)/\sqrt{2\pi}$ and $h > 0$ is the kernel bandwidth. Consider further a definite integral over the KDE above:

$$P(a, b) := \int_b^a \widehat{f}_h(x) dx. \tag{4}$$

If we replace the empirical c.d.f. in Eq. (2) with the KDE c.d.f. using Eq. (4), we obtain the following transformation:

$$\hat{F}_N^{\text{KDQ,naive}}(x) := P(-\infty, x). \tag{5}$$

However, we slightly modify the above definition to match the behavior of the min-max transformation and the quantile transformation at the extrema $X_{(1)}$ and $X_{(N)}$, such that we obtain $\hat{F}_N^{\text{KDQ}}(x) = 0$ for $x \leq X_{(1)}$ and $\hat{F}_N^{\text{KDQ}}(x) = 1$ for $X_{(N)} \leq x$. To do this, while keeping $\hat{F}_N^{\text{KDQ}}(x)$ continuous, we define the final version of the kernel density quantile (KD-quantile) transformation as follows:

$$\hat{F}_N^{\text{KDQ}}(x) := \begin{cases} \frac{P(X_{(1)},x)}{P(X_{(1)},X_{(N)})}, & X_{(1)} \leq x \leq X_{(N)} \\ 0, & x \leq X_{(1)} \\ 1, & X_{(N)} \leq x. \end{cases} \tag{6}$$

This is equivalent to removing probability mass from the KDE outside the range $[X_{(1)}, X_{(N)}]$. In practice, we also parameterize $h = \alpha\hat{\sigma}_X^2$, where α is the *bandwidth factor*, and $\hat{\sigma}_X^2$ is an estimate of the variance of X.

As the kernel bandwidth $h \to \infty$, the KD-quantile transformation will converge towards the min-max transformation. Meanwhile, when $h \to 0$, the KD-quantile transformation converges towards the vanilla quantile transformation. Furthermore, as noted previously, $\hat{F}_N^{\text{KDQ,naive}}(x)$ is exactly the formula for computing the kernel density estimator of quantiles. However, in this paper, we propose (for the first time, to our knowledge) choosing a large kernel bandwidth. Rather than estimating quantiles with improved statistical efficiency as in (Sheather, 1990; Kulczycki & DaWidowicz, 1999), our aim is carry out a transformation that is an optimal compromise between the min-max transformation and the quantile transformation, for a given downstream task. As we will see in the experiment section, this optimal compromise is often struck at large bandwidths such as $h = 1 \cdot \hat{\sigma}_X^2$, even as the sample size $N \to \infty$.

Efficient computation As a supervised learning preprocessing method, KD-quantile transformation is intended to be used in the context of separate train and test sets. Thus, it is desirable that the transformation be estimated quickly on a training set, efficiently (yet approximately) represented with a fixed number of reference points even as $N \to \infty$, and then applied with low runtime cost to new incoming samples. To do this, for each feature we take $M = \min(10000, N)$ samples without replacement, and use this subsample to form the KDE in Eq. (3) and to compute definite integrals in Eq (4). Having obtained M definite integrals, we then compute KD-quantiles at $Q = \min(1000, M)$ reference points, equally spaced in $[0, 1]$. New points are transformed by linearly interpolating these KD-quantiles, analogously to the method for computing vanilla quantiles via linear interpolation of the inverse of the empirical cumulative distribution function. Software is available at `https://github.com/uhhnonymous/anon-tmlr-2023-08`.

2.3 Application to correlation analysis

In correlation analysis, whereas Pearson's r (Pearson, 1895) is appropriate for measuring the strength of linear relationships, Spearman's rank-correlation coefficient ρ (Spearman, 1904) is useful for measuring the strength of non-linear yet monotonic relationships. Spearman's ρ may be computed by applying Pearson's correlation coefficient after first transforming the original data to quantiles or ranks $R(x) := N\hat{F}_N(x)$. Thus, it is straightforward to extend Spearman's ρ by computing the correlation coefficient between two variables by computing their respective KD-quantiles, then applying Pearson's formula as before. Like Spearman's ρ, it is apparent that ours is a particular case of a general correlation coefficient Γ Kendall (1948). For N samples of random variables X and Y, the general correlation coefficient Γ may be written as

$$\Gamma = \frac{\sum_{i,j=1}^{N} a_{ij} b_{ij}}{\sqrt{\sum_{i,j=1}^{N} a_{ij}^2 \sum_{i,j=1}^{N} b_{ij}^2}},$$

for $a_{ij} := r_j - r_i, b_{ij} := s_j - s_i$, where now r_i and s_i correspond to the KD-quantiles of X_i and Y_i, respectively.

Because ranks are robust to the effect of outliers, Spearman's ρ is also useful as a robust measure of correlation; our proposed approach inherits this benefit, as will be shown in the experiments.

2.4 Application to univariate clustering

Here we apply KD-quantile transformation to the problem of univariate clustering (a.k.a. discretization). Our approach relies on the intuition that local minima and local maxima of the KDE will tend to correspond to cluster boundaries and cluster centroids, respectively. However, naive application of this idea would perform poorly because low-density regions will tend to have many isolated extrema, causing us to partition low-density regions into many separate clusters. When we apply the KD-quantile transformation, we draw such points in low-density regions closer together, because the definite integrals between such points will tend to be small. Then, when we form a KDE on these transformed points and identify local extrema, we avoid partitioning such low-density regions into many separate clusters.

Our proposed approach thus comprises three steps:

1. Compute $T_n = \hat{F}_N^{\mathrm{KDQ}}(X_n),\ \forall n \in \{1, \ldots, N\}$.
2. Form the kernel density estimator $\hat{f}_h(t)$ for $T_1, \ldots, T_N$. For this second KDE, select a vanishing bandwidth via Scott's Rule (Scott, 1992).
3. Identify the cluster boundaries from the local minima in $\hat{f}_h(t)$, and inverse-KD-quantile-transform the boundaries for T_n to obtain boundaries for clustering X_n.

3 Experiments

In this section, we evaluate our approach on supervised classification problems with real-world tabular datasets, on correlation analyses using simulated and real data, and on clustering of simulated univariate datasets with known ground-truth.

3.1 Feature preprocessing for supervised learning

3.1.1 Classification with PCA and Gaussian Naive Bayes

We first replicate the experimental setup of (Raschka, 2014), analyzing the effect of feature preprocessing methods on a simple classifier for the Wine dataset (Forina *et al.*, 1988). The classification pipeline comprises feature preprocessing, principle component analysis (PCA) with 2 components, followed by a Gaussian Naive Bayes classifier. In addition to z-score standardization and min-max scaling, used in (Raschka, 2014), we also try quantile transformation and our proposed KD-quantile transformation approach. For the KD-quantile transformation, we show results both for the default bandwidth factor of 1 (i.e. $h = 1 \cdot \sigma^2$) and for a sweep of bandwidth factors between 0.1 and 10.

The accuracy, averaged over 100 simulated train-test splits, is shown in Figure 1(A). For a small bandwidth factor, KD-quantile transformation performs as poorly as quantile transformation, while for a large bandwidth factor, it approaches the accuracy of min-max scaling. However, for intermediate bandwidths, and at the default setting in particular, our approach offers a superior "happy medium".

In Figure 2, we illustrate the effect of min-max scaling, quantile transformation, and KD-quantile transformation on the `MalicAcid` feature in the Wine dataset. We see that KD-quantiles have reduced the distance of outliers compared to the original data, while preserving the overall bimodal shape of the feature distribution.

We repeat the above experimental setup for three more standard classification datasets: Iris (Fisher, 1936) shown in Figure 1(B), Penguins (Gorman *et al.*, 2014) in Figure 1(C), and Hawks (Cannon *et al.*, 2019) in Figure 1(D). In all cases, KD-quantile transformation can be tuned to outperform both min-max scaling and quantile transformation, and KD-quantile using the default bandwidth factor is non-inferior to the better of the two. KD-quantile also outperformed z-score scaling, except in the case of the Penguins dataset.

3.1.2 Linear classification on Small Data Benchmarks

We next compared preprocessing methods on a dataset-of-datasets benchmark, comprising 142 tabular datasets, each with at 50 samples. We replicated the experimental setup of the Small Data Benchmarks (Feldman, 2021) on the UCI++ dataset repository (Paulo *et al.*, 2015). In (Feldman, 2021), the leading linear classifier was a support vector classifier (SVC) with min-max preprocessing, to which we appended SVC with quantile transformation and SVC with KD-quantile transformation. For the latter, we only used the default bandwidth factor $\alpha = 1$, so as to give equal hyperparameter optimization budgets to each approach. As in (Feldman, 2021), for each preprocessing method, we optimized the regularization hyperparameter $C \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$, evaluating each method via one-vs-rest-weighted ROC AUC, averaged over 4 stratified cross-validation folds.

Our results are summarized in Table 1. We see that KD-quantile transformation provides greater average ROC AUC, with less variance, at the same tuning budget as the other approaches. We further analyzed the

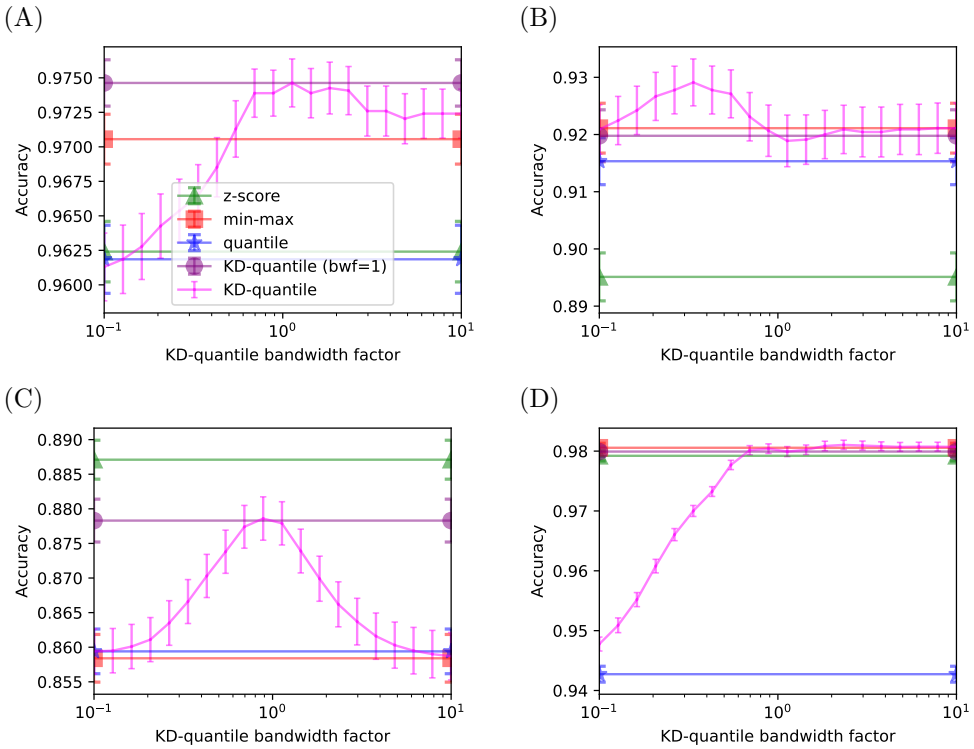

Figure 1: Accuracy on four tabular classification problems for different feature preprocessing methods. Results are shown for Wine (A), Iris (B), Penguins (C), and Hawks (D). Accuracy is shown as a horizontal line for z-scaling, min-max scaling, quantile transformation, and KD-quantile transformation with the default bandwidth factor $\alpha = 1$. In magenta, we show accuracy as a function of KD-quantile bandwidth factor α.

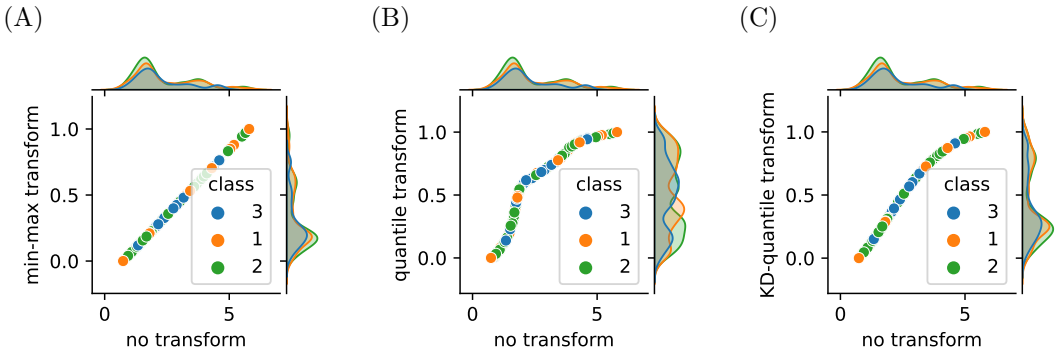

Figure 2: Comparison of (A) min-max scaling, (B) quantile transformation, and (C) KD-quantile transformation on the `MalicAcid` feature in the Wine dataset. The horizontal density plots at the top of each of the subplots depict the distribution of the original data, while the vertical density plots to the right of each of the subplots show the distribution after each preprocessing step. The scatterplots within each subplot reflect the fact that all preprocessing transforms are monotonic.

Table 1: Performance of preprocessing methods on Small Data Benchmarks, as measured by area under the Receiver Operating Characteristic curve (ROC AUC). The columns display the mean and standard deviation of the ROC AUC, computed over 142 datasets in the benchmark.

METHOD	Mean ROC AUC	StdDev ROC AUC
Min-max scaling	0.864	0.132
Quantile transformation	0.866	0.131
KD-quantile transformation	0.868	0.129

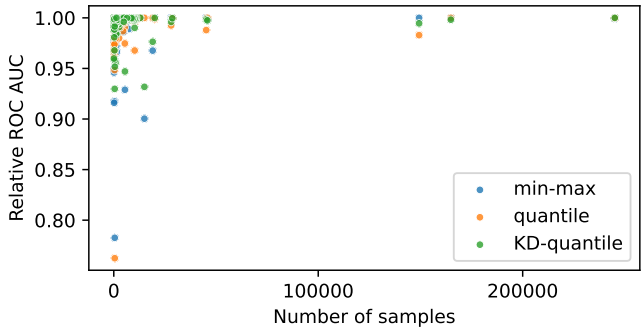

Figure 3: Performance of preprocessing methods on Small Data Benchmarks, plotted against dataset size. For each dataset, we compute the relative ROC AUC for a given method as its own ROC AUC, divided by the maximum ROC AUC over all three preprocessing methods for that dataset.

performance of the different methods in terms of the number of samples in each dataset in Figure 3. Plotting the relative ROC AUC against the number of samples N, we see that our proposed approach is particularly helpful in avoiding suboptimal performance for small-N datasets.

3.2 Correlation analysis

To provide a basic intuition, we first illustrate the different methods on synthetic datasets shown in Figure 4, replicating the example from (Wikipedia, 2009). When two variables are monotonically but not linearly related, as in Figure 4(A), the Spearman correlation exceeds the Pearson correlation. In this case, our approach behaves similarly to Spearman's. When two variables have noisy linear relationship, as in Figure 4(B), both the Pearson and Spearman have moderate correlation, and our approach interpolates between the two. When two variables have a linear relationship, yet are corrupted by outliers, the Pearson correlation is reduced due to the outliers, while the Spearman correlation is robust to this effect. In this case, our approach also behaves similarly to Spearman's.

Next, we perform correlation analysis on the California housing dataset (Pace & Barry, 1997), containing district-level features such as average prices and number of bedrooms from the 1990 Census. Overall, the computed correlation coefficients are typically close, with only a few exceptions, as shown in Figure 5. Our approach's top two disagreements with Pearson's are on (`AveBedrms`, `AveRooms`) and (`MedInc`, `AveRooms`). Our approach's top two disagreements with Spearman's are on (`AveBedrms`, `AveRooms`) and (`Population`, `AveBedrms`). We further observe that KD-quantile-based correlations typically, but not always, lie between the Pearson and Spearman correlation coefficients.

We analyze the correlation disagreements for (`AveBedrms`, `AveRooms`) in Figure 6. From the original data, it is apparent that `AveBedrms` and `AveRooms` are correlated, whether we examine

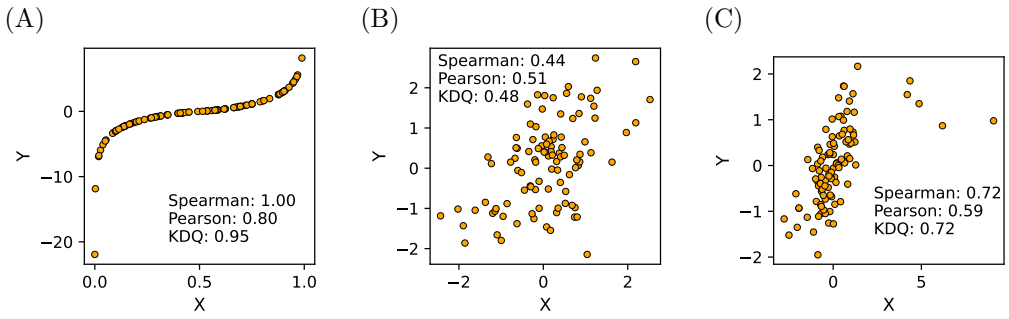

Figure 4: Illustration of correlation analysis using Pearson's r, Spearman's ρ, and our proposed approach, for simulated data. Three scenarios are depicted: (A) nonlinear yet monotonic relationship, (B) noisy linear relationship, and (C) linear relationship corrupted by outliers.

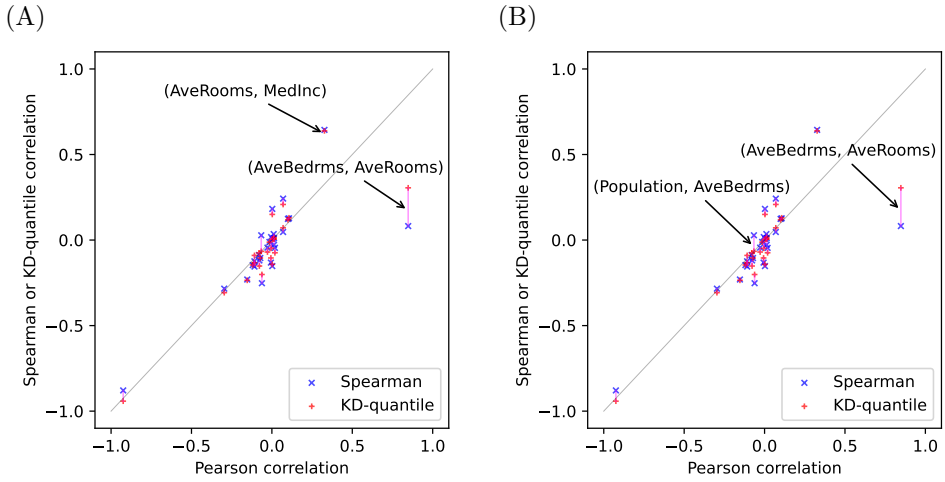

Figure 5: Correlation coefficients derived from the California housing dataset. The pink line depicts the gap between Spearman and KD-quantile correlations, while the distance from the gray line shows how far each are from the Pearson correlation. Both (A) and (B) contain the same data, but top disagreements between ours and Pearson's, and ours and Spearman's are highlighted separately in (A) and (B), respectively.

the full dataset or exclude outlier districts. This relationship was obscured by quantile transformation (and thus, by Spearman correlation analysis), whereas it is still noticeable after KD-quantile transformation.

We repeat the analysis of disagreement for (`MedianIncome`, `AverageRooms`) and (`Population`, `AverageBedrooms`) in Figure 7. For the former disagreement, our approach agrees with quantile transformation-based analysis, identifying the typical positive dependence between median income and average rooms, by reducing the impact of districts with extremely high average rooms. For the latter disagreement, our approach agrees with original data-based analysis, identifying the negative relationship one would expect to observe between district population (and therefore density) and the average number of bedrooms.

3.3 Clustering univariate data

In this experiment, we generate five separate synthetic univariate datasets, each sampled according to the following mixture distributions:

- $0.55 * \mathcal{N}(\mu = 1, \sigma = 0.75) + 0.30 * \mathcal{N}(\mu = 4, \sigma = 1) + 0.15 * \text{Unif}[a = 0, b = 20]$
- $0.45 * \mathcal{N}(\mu = 1, \sigma = 0.5) + 0.45 * \mathcal{N}(\mu = 4, \sigma = 1) + 0.10 * \text{Unif}[a = 0, b = 20]$
- $0.67 * \mathcal{N}(\mu = 1, \sigma = 0.5) + 0.33 * \mathcal{N}(\mu = 4, \sigma = 1)$
- $0.8 * \text{Exp}(\lambda = 1) + 0.2 * [10 + \text{Exp}(\lambda = 4)]$
- $0.5 * \text{Exp}(\lambda = 8) + 0.5 * [100 - \text{Exp}(\lambda = 5)]$.

We compare our approach to five other clustering algorithms: KMeans with K chosen to maximize the Silhouette Coefficient (Rousseeuw, 1987), GMM with K chosen via the Bayesian information criterion (BIC), Bayesian Gaussian Mixture Model (GMM) with a Dirichlet Process prior, Mean Shift clustering (Comaniciu & Meer, 2002), and HDBSCAN (Campello *et al.*, 2013; McInnes *et al.*, 2017) (with `min_cluster_size=5, cluster_selection_epsilon=0.5`).

In Figure 8, we plot stacked histograms to compare the ground-truth cluster identities of the data with the estimated cluster identities from each of the methods, for $N = 500$ samples. On the top row, we depict the true clusters, as well as the true number of mixture components. On each of the following rows, we show the distributions of estimated clusters for each the methods. We see that our approach is the only method to correctly infer the true number of mixture components, and that it partitions the space similarly to ground-truth. GMM with BIC performed well on the first three datasets, but not on the last two. Meanwhile, KMeans performed well on the last three datasets but not on the first two. Bayesian GMM tended to slightly overestimate the number of components, while MeanShift and HDBSCAN tended to aggressively overestimate.

We also include, on the penultimate row above our proposed approach, results for an ablation of our proposed approach, in which we define separate clusters at the local minima of the KDE of unpreprocessed inputs, rather than on the KDE of KD-quantiles. We see that the ablated method fails on the first, second, and fourth datasets, where there is a large imbalance between the mixture weights of the cluster components.

We repeated the above experimental setup, this time varying the number of samples $N \in \{100, 200, 500, 1000, 2000, 5000\}$, and performing 20 independent simulations per each setting of N. For each simulation, we recorded whether the true K and estimated $\hat{K}$ number of components matched, as well as the the adjusted Rand index (ARI) between the ground-truth and estimated cluster labelings, for each method. The results, averaged over 20 simulations, are shown in Figure 9. Our approach is the only method to attain high accuracy across all settings when $N > 1000$. Alternative methods behave inconsistently across datasets or across varying sample sizes. For the first two datasets with small N, KD-quantile is outperformed by GMM and Bayesian GMM. However, GMM struggles on the first two datasets for large N, and on the fourth and fifth datasets; meanwhile, Bayesian GMM struggles on all datasets for large N.

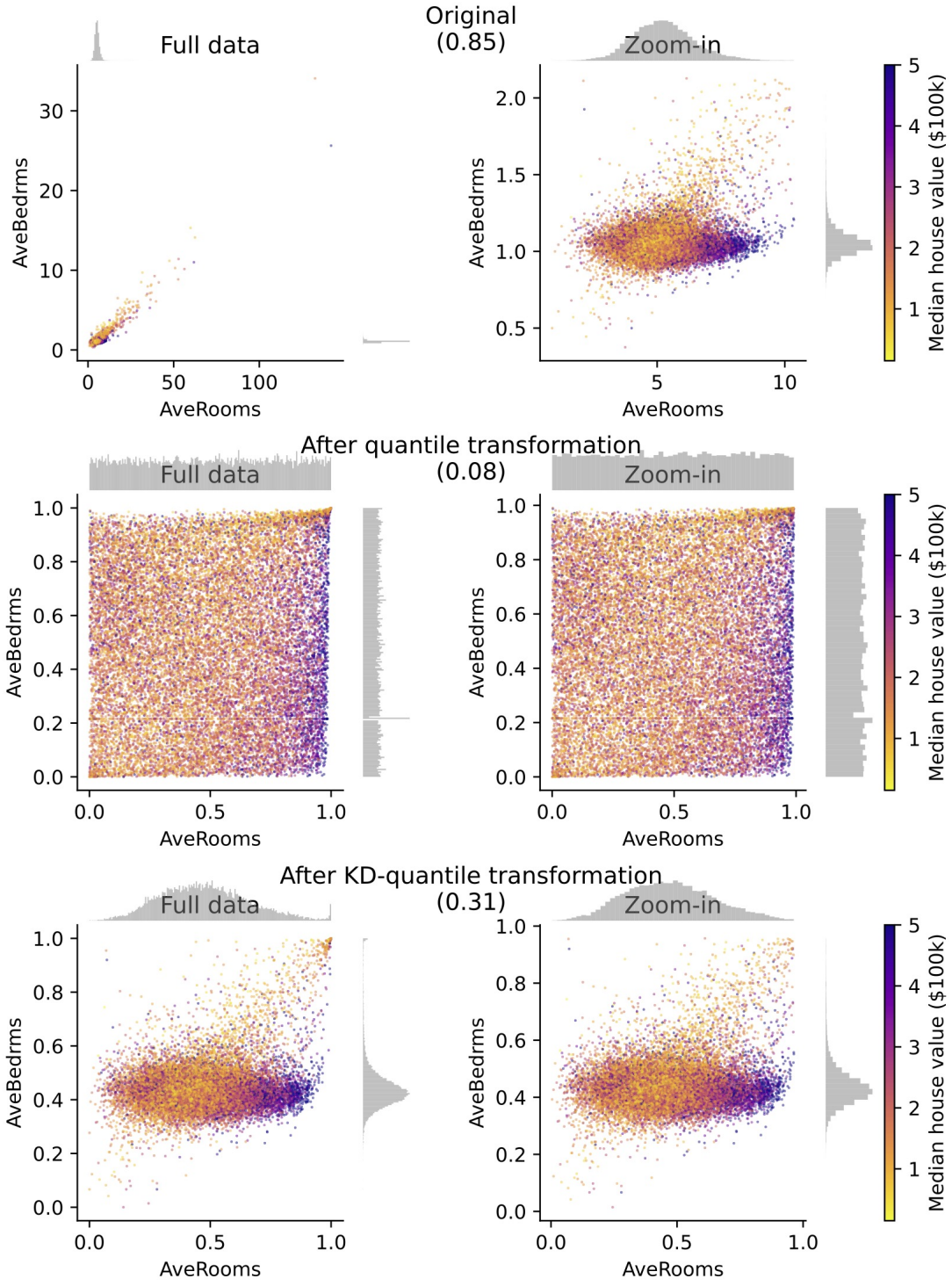

Figure 6: Correlation analysis for (`AverageBedrooms`, `AverageRooms`) in the California housing dataset. The rows, from top to bottom, correspond to original data, quantile transformation, and KD-quantile transformation. The full dataset is shown on the left, while outliers are excluded on the right. Each district is colored by its median house value. In parenthesis above each row are the Pearson (0.85), Spearman (0.08), and KD-quantile (0.31) estimated correlations between the variables.

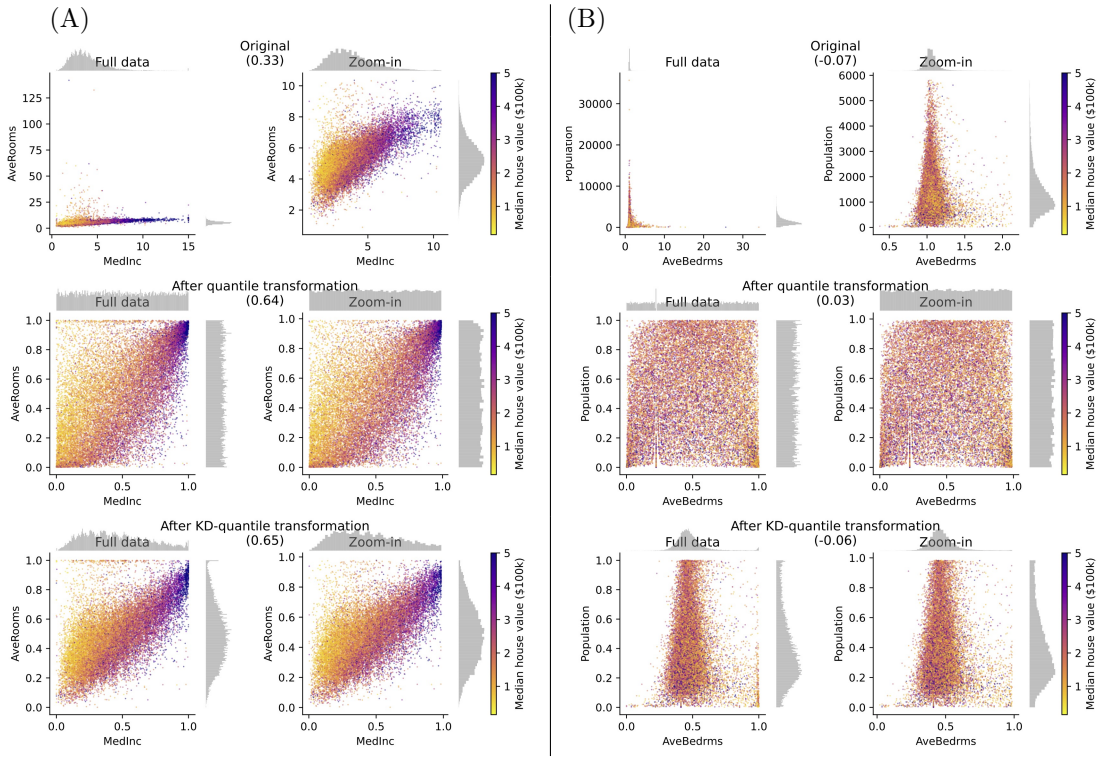

Figure 7: Correlation analysis for (`MedianIncome`, `AverageRooms`) (A) and (`Population`, `AverageBedrooms`) (B) in the California housing dataset. See Figure 6 for an explanation of the plot.

4 Related Work

As far as we know, the use of kernel density quantiles as a compromise between min-max scaling and quantile transformation has not been previously proposed in the literature. Similarly, we are not aware of kernel-density quantiles being proposed to balance between the strengths of Pearson's r and Spearman's ρ.

Our proposed approach for univariate clustering is similar in spirit to various density-based clustering methods, including mean-shift clustering (Comaniciu & Meer, 2002), level-set trees (Schmidberger & Frank, 2005; Wang & Huang, 2009; Kent *et al.*, 2013), and HDBSCAN (Campello *et al.*, 2013). However, such methods tend to leave isolated points as singletons, while joining points in high-density regions into larger clusters. To our knowledge, our approach for compressing together such isolated points has not been previously considered.

The kernel density estimator (KDE) was previously proposed (Flores *et al.*, 2019) in the context of discretization-based preprocessing for supervised learning. However, their method did not use kernel-density quantiles as a preprocessing step, but instead employed a supervised approach that, for a multiclass classification problem with C classes, constructed C different KDEs for each feature.

5 Discussion

Limitations Our approach for supervised preprocessing is limited by the fact that, even though the bandwidth factor is a tunable continuous parameter, it cannot be directly optimized but needs to be chosen via hyperparameter tuning. Meanwhile, our proposed approaches for correlation analysis and univariate clustering would benefit from further theoretical analysis. Furthermore, it is not clear whether there may exist multivariate generalizations of the KD-quantile transformation which would apply to multivariate clustering.

Future Work In this paper, we have focused on per-feature transformation, in which each feature is mapped to quantiles computed across all samples for a given feature. But, especially in genomics, it is

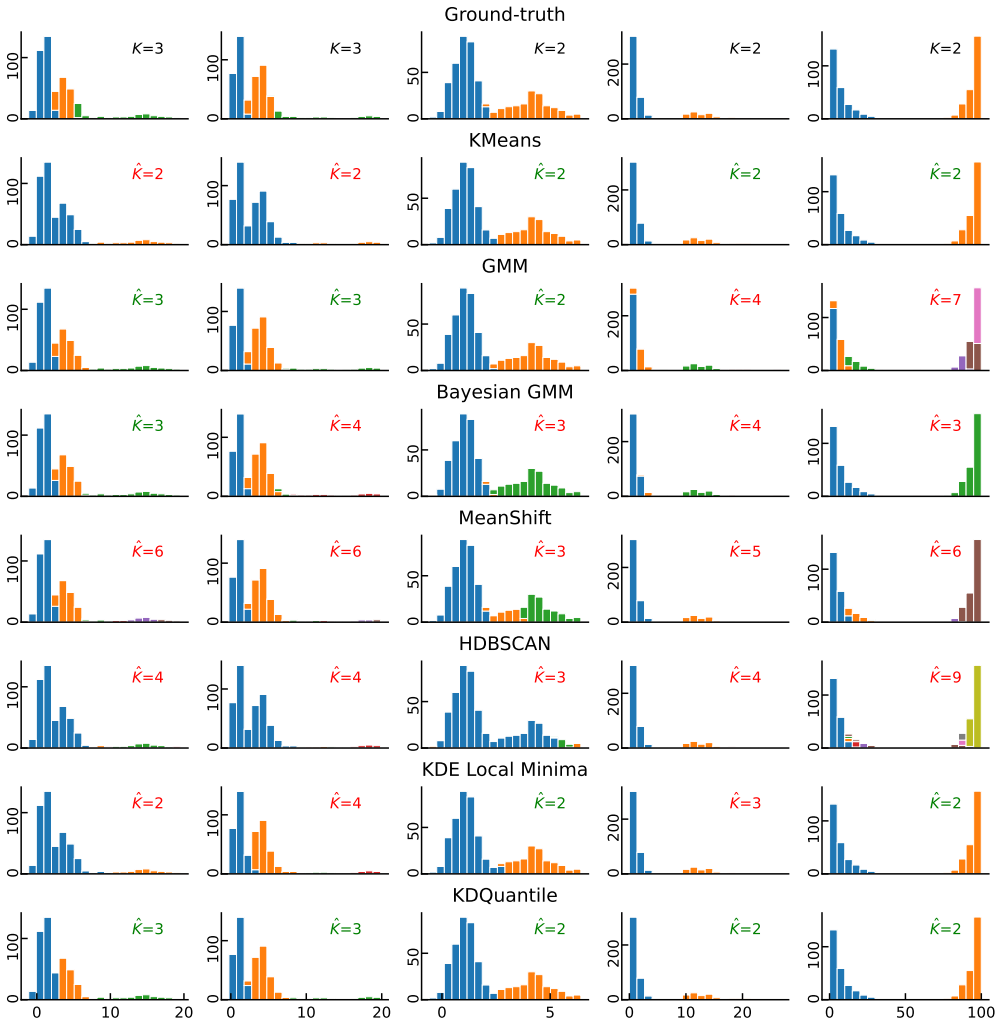

Figure 8: Clustering performance for five datasets (one dataset per column), generated with $N = 500$ samples. On each of the rows, for the different methods, we indicate the estimated number of components (green if correct, red if incorrect).

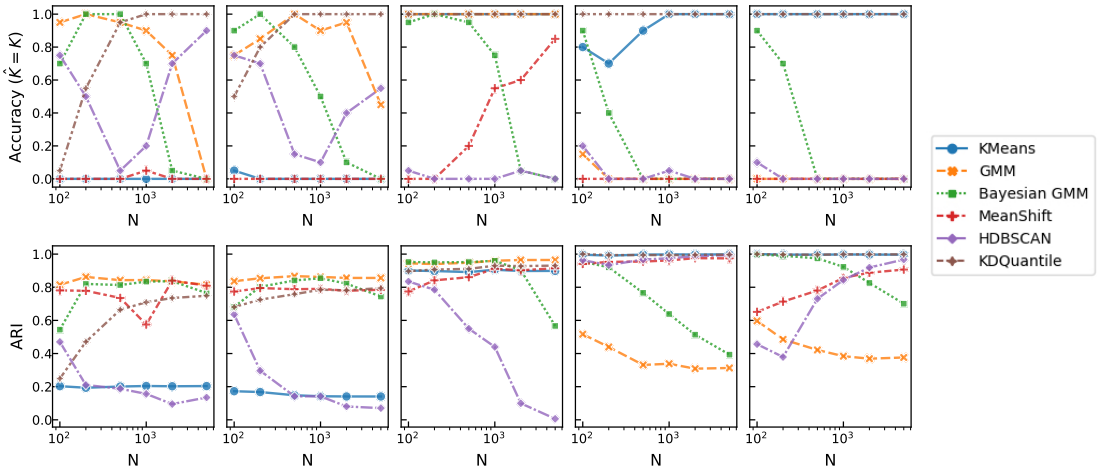

Figure 9: Performance at clustering univariate features, for varying number of samples N. The top row depicts the fraction of simulations in which the estimated number of mixture components equaled the true number. The bottom row depicts the average ARI between the ground-truth clustering and the estimated clusters. Each of the five columns corresponds to five mixture distributions described in the text and depicted in Figure 8.

common to perform per-sample quantile normalization (Bolstad *et al.*, 2003; Amaratunga & Cabrera, 2001), in which features for a sample are mapped to quantiles computed across all the features for that sample. It would be profitable to examine the use of KD-quantile transformation in this other context.

The use of quantiles is not limited to classic tabular machine learning problems. For example, quantile regression (Koenker & Bassett Jr, 1978) has recently found increasing use in conformal prediction (Romano *et al.*, 2019; Liu *et al.*, 2022), uncertainty quantification (Jeon *et al.*, 2016), and reinforcement learning (Rowland *et al.*, 2023). Future work could study whether using kernel-density quantiles could be profitably used in place of vanilla quantiles in such settings.

6 Conclusion

In this paper, we proposed the use of the kernel-density quantile transformation as a nonlinear preprocessing step, both for supervised learning settings and for statistical data analysis. In a variety of experiments on simulated and real datasets, we demonstrated that our proposed approach is straightforward to use, requiring simple (or no) tuning to offer improved performance compared to previous approaches.

References

Amaratunga, Dhammika, & Cabrera, Javier. 2001. Analysis of data from viral DNA microchips. *Journal of the American Statistical Association*, **96**(456), 1161–1170.

Bartlett, Maurice S. 1947. The use of transformations. *Biometrics*, **3**(1), 39–52.

Bolstad, Benjamin M, Irizarry, Rafael A, Åstrand, Magnus, & Speed, Terence P. 2003. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, **19**(2), 185–193.

Campello, Ricardo JGB, Moulavi, Davoud, & Sander, Jörg. 2013. Density-based clustering based on hierarchical density estimates. *Pages 160–172 of: Pacific-Asia conference on knowledge discovery and data mining*. Springer.

Cannon, A, Cobb, G, Hartlaub, B, Legler, J, Lock, R, Moore, T, Rossman, A, & Witmer, J. 2019. Stat2Data: Datasets for Stat2. *R package version*, **2**(0).

Comaniciu, Dorin, & Meer, Peter. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, **24**(5), 603–619.

Feldman, Sergey. 2021. *Which machine learning classifiers are best for small datasets? an empirical study.* `https://github.com/sergeyf/SmallDataBenchmarks`.

Fisher, Ronald A. 1936. The use of multiple measurements in taxonomic problems. *AnnEug*, **7**(2), 179–188.

Flores, Jose Luis, Calvo, Borja, & Perez, Aritz. 2019. Supervised non-parametric discretization based on Kernel density estimation. *Pattern Recognition Letters*, **128**, 496–504.

Forina, Michele, Leardi, Riccardo, Armanino, C, Lanteri, Sergio, Conti, Paolo, Princi, P, *et al.* 1988. PARVUS an extendable package of programs for data exploration, classification and correlation. *Elsevier*.

Gorman, Kristen B, Williams, Tony D, & Fraser, William R. 2014. Ecological sexual dimorphism and environmental variability within a community of Antarctic penguins (genus Pygoscelis). *PloS one*, **9**(3), e90081.

Jeon, Soyoung, Paciorek, Christopher J, & Wehner, Michael F. 2016. Quantile-based bias correction and uncertainty quantification of extreme event attribution statements. *Weather and Climate Extremes*, **12**, 24–32.

Kendall, Maurice George. 1948. *Rank correlation methods.* Griffin.

Kent, Brian P, Rinaldo, Alessandro, & Verstynen, Timothy. 2013. Debacl: A python package for interactive density-based clustering. *arXiv preprint arXiv:1307.8136*.

Koenker, Roger, & Bassett Jr, Gilbert. 1978. Regression quantiles. *Econometrica: journal of the Econometric Society*, 33–50.

Kulczycki, Piotr, & DaWidowicz, Antoni Leon. 1999. Kernel estimator of quantile. *ZESZYTY NAUKOWE-UNIWERSYTETU JAGIELLONSKIEGO-ALL SERIES-*, **1236**, 325–336.

Liu, Meichen, Ding, Lei, Yu, Dengdeng, Liu, Wulong, Kong, Linglong, & Jiang, Bei. 2022. Conformalized Fairness via Quantile Regression. *Advances in Neural Information Processing Systems*, **35**, 11561–11572.

McInnes, Leland, Healy, John, & Astels, Steve. 2017. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, **2**(11), 205.

Pace, R Kelley, & Barry, Ronald. 1997. Sparse spatial autoregressions. *Statistics & Probability Letters*, **33**(3), 291–297.

Parzen, Emanuel. 1962. On estimation of a probability density function and mode. *The annals of mathematical statistics*, **33**(3), 1065–1076.

Paulo, Luis, dos Santos, Davi Pereira, & Horta, Danilo. 2015 (Jan.). *ucipp 1.1*.

Pearson, Karl. 1895. VII. Note on regression and inheritance in the case of two parents. *proceedings of the royal society of London*, **58**(347-352), 240–242.

Raschka, S. 2014. *About feature scaling and normalization–and the effect of standardization for machine learning algorithms. Sebastian Raschka. 2014*.

Romano, Yaniv, Patterson, Evan, & Candes, Emmanuel. 2019. Conformalized quantile regression. *Advances in neural information processing systems*, **32**.

Rosenblatt, Murray. 1956. Remarks on some nonparametric estimates of a density function. *The annals of mathematical statistics*, 832–837.

Rousseeuw, Peter J. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, **20**, 53–65.

Rowland, Mark, Munos, Rémi, Azar, Mohammad Gheshlaghi, Tang, Yunhao, Ostrovski, Georg, Harutyunyan, Anna, Tuyls, Karl, Bellemare, Marc G, & Dabney, Will. 2023. An analysis of quantile temporal-difference learning. *arXiv preprint arXiv:2301.04462*.

Schmidberger, Gabi, & Frank, Eibe. 2005. Unsupervised discretization using tree-based density estimation. *Pages 240–251 of: European conference on principles of data mining and knowledge discovery*. Springer.

Scott, DW. 1992. Multivariate Density Estimation. *Multivariate Density Estimation*.

Sheather, Simon J. 1990. Kernel quantile estimators. *Journal of the American Statistical Association*, **85**(410), 410–416.

Silverman, BW. 1986. Density estimation for statistics and data analysis. *Monographs on Statistics and Applied Probability*.

Spearman, C. 1904. The proof and measurement of association between two things. *Am J Psychol*, **15**, 72–101.

Van der Waerden, BL. 1952. Order tests for the two-sample problem and their power. *Pages 453–458 of: Indagationes Mathematicae (Proceedings)*, vol. 55. Elsevier.

Wang, Xiao-Feng, & Huang, De-Shuang. 2009. A novel density-based clustering framework by using level set method. *IEEE Transactions on knowledge and data engineering*, **21**(11), 1515–1531.

Wikipedia. 2009. *Spearman's rank correlation coefficient*. [Online; accessed 8-August-2023].

Yamato, Hajime. 1973. Uniform convergence of an estimator of a distribution function. *Bulletin of Mathematical Statistics*.