Kernel bandwidth estimation for non-parametric density estimation: a comparative study

Christiaan M. van der Walt
Modelling and Digital Science, CSIR, Pretoria 0001
Multilingual Speech Technologies Group,
North-West University,
Vanderbijlpark 1900, South Africa
cvdwalt@csir.co.za

Etienne Barnard
Multilingual Speech Technologies Group
North-West University
Vanderbijlpark 1900, South Africa
etienne.barnard@nwu.ac.za

***Abstract*—We investigate the performance of conventional bandwidth estimators for non-parametric kernel density estimation on a number of representative pattern-recognition tasks, to gain a better understanding of the behaviour of these estimators in high-dimensional spaces. We show that there are several regularities in the relative performance of conventional kernel bandwidth estimators across different tasks and dimensionalities. In particular, we find that the Silverman rule-of-thumb and maximal-smoothing principle estimators consistently perform competitively on most tasks and dimensions for the datasets considered.**

Keywords—non-parametric density estimation; kernel density estimation; kernel bandwidth estimation, pattern recognition

I. KERNEL DENSITY ESTIMATION

Kernel Density Estimators (KDEs) estimate the non-parametric density function of a set of *D*-dimensional *iid* data samples, ***X***, as the sum of parametric functions, where the parametric function is known as a kernel and is centred on each sample. More formally, the density of a data point **x**, can be estimated as

$$\hat{p}_{\mathbf{H}}(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N} K\left(\mathbf{x} - \mathbf{x}_j \mid \mathbf{H}_j\right) \qquad (1)$$

where N is the number of training samples, $\mathbf{x}_j$ is the j'th sample in the dataset $\mathbf{X}$, K is the kernel function (typically a parametric density function such as the Gaussian distribution) and $\mathbf{H}_j$ is the bandwidth matrix that describes the variance of the kernel function centred on $\mathbf{x}_j$.

KDEs thus require the selection of two design parameters, namely the parametric form of the kernel function and the bandwidth matrix. It has been shown that the efficiencies of kernels with respect to the Mean Squared Error (MSE) between the true and estimated distribution do not differ significantly, and that the choice of kernel function should rather be based on the mathematical properties of the kernel function, since the estimated density function inherits the smoothness properties of the kernel function [1]. The Gaussian kernel is therefore often selected in practice for its smoothness properties, such as continuity and differentiability. This thus leaves the estimation of the kernel bandwidth as the only parameter to be estimated.

Since the introduction of KDEs in 1958 [2], several bandwidth estimation algorithms for kernel density estimation have been proposed. These bandwidth estimators were mainly developed in the field of statistics, and were not intended for density estimation in high-dimensional spaces – as are often encountered in pattern recognition problems. Scott, for example, states that density estimation beyond 6 dimensions with conventional approaches is often regarded as practically infeasible [3]. We therefore define high-dimensional data as data with dimensionalities of 10 and higher.

Recent advances in pattern-recognition have shown that kernel density estimation for high-dimensional pattern-recognition tasks is indeed possible [4, 5] by making use of the maximum-likelihood (ML) leave-one-out (LOUT) framework, and that these estimators outperform conventional kernel bandwidth estimators in practice. ML bandwidth estimators are however, dependent on the estimation of initial bandwidth to initialise the bandwidth optimisation procedure. We therefore investigate the performance of conventional bandwidth estimation algorithms on a representative set of pattern recognition tasks, to (1) gain a better understanding of their behaviour over a representative range of dimensionalities and number of samples, and (2) to determine if conventional bandwidth estimators can successfully be used to initialise the bandwidth optimisation procedures of the above mentioned ML estimators, specifically in high-dimensional spaces.

In the next section we give a brief summary of the most notable conventional kernel bandwidth estimators that will be compared, and the pattern-recognition tasks that will be investigated in this study.

II. METHODS AND DATA

A. Conventional kernel bandwidth estimators

Conventional kernel bandwidth estimators can be broadly categorised into rule-of-thumb, least-squares cross-validation (LSCV), likelihood CV (LCV), and plugin methods. We briefly summarise conventional kernel bandwidth estimators that fall within these categories and only present the univariate case, since all univariate estimators can be extended to the multivariate case by simply estimating bandwidths for each

variable independently – thus, assuming independence between variables.

Rule of thumb estimators typically optimise the Asymptotic Mean Integrated Squared Error (AMISE) with respect to the kernel bandwidth. The expression of the root of the derivatie of the AMISE with respect to the bandwidth unfortunately relies on the unknown distribution, *f(x)*, and therefore an assumptions regarding the distribution must be made. The *Silverman rule-of-thumb* estimator assumes a Gaussian distribution as reference distribution for *f(x)*, and if a Gaussian kernel is assumed, the optimal bandwidth is derived as

$$h_{Silv} = 1.06\hat{\sigma}N^{-\frac{1}{5}} \tag{2}$$

where $\hat{\sigma}^2$ is the sample variance and N is the number of samples

The Maximal Smoothing Principle (MSP) approach selects a reference distribution for *f(x)* that minimises the "roughness" of the density, defined as

$$R(f'') = \int \left(f(x)''\right)^2 \tag{3}$$

This yields the upper bound of the optimal AMISE bandwidth, which gives the maximal smoothing compatible with the scale of the density estimate [6]. It has been shown [7] that the beta distribution minimises the "roughness" if the variance is used as scale parameter, and if a Gaussian kernel is assumed the bandwidth that maximises the smoothing of the density estimate for a given scale is derived as

$$h_{MSP} = 1.144\hat{\sigma}N^{-\frac{1}{5}} \tag{3}$$

LSCV approaches attempt to minimise the Integrated Squared Error (ISE) of the density estimate with respect to the kernel bandwidth. By using a method of moments estimate, the LSCV objective function is estimated as

$$LSCV(h) = \int \left(\hat{f}_h(y)\right)^2 dy - 2\sum_{i=1}^{N} \hat{f}_{-i,h}(x_i), \tag{4}$$

where $\hat{f}_{-i,h}(x)$ denotes the leave-one-out kernel density estimate with bandwidth *h* for data point *x*, when data point x_i is left out. Optimising the bandwidth with respect to this objective function thus requires a numerical optimisation procedure. In practice the optimisation problem becomes ill conditioned and practically infeasible if a unique bandwidth is estimated for each kernel, thus an identical bandwidth is typically estimated for all kernels in practice.

LCV approaches attempt to maximize the LOUT ML criteria with respect to the kernel bandwidth. Since the maximisation of the ML criteria has a trivial solution at *h=0*, LOUT CV is employed to prevent this degenerate solution. The LOUT ML objective function is defined as

$$LCV(h) = \frac{1}{N}\sum_{i=1}^{N} \log\left(\hat{f}_{-i,h}(x_i)\right). \tag{4}$$

If this objective function is optimised with respect to a single bandwidth, the procedure is known as *Uniform LCV* (ULCV).

To estimate a unique bandwidth per kernel, the ULCV bandwidth can be adapted locally with the nearest-neighbour distance. Scaling with the nearest-neighbour distance prevents to optimisation of numerous bandwidths simultaneously, and makes the estimation of a unique bandwidth per kernel practically feasible. This estimator is known as the *Local LCV* (LLCV) estimator.

Plugin methods attempt to find a closed-form solution for the optimisation of the AMISE with respect to the bandwidth. Similar to rule-of-thumb approaches, calculation of the optimal bandwidth relies on the unknown distribution *f*. Instead of assuming a reference distribution (as for the rule-of-thumb approaches) for the unknown density, *f*, a pilot estimate of the unknown density is calculated. The pilot density also, however, requires the estimate of a bandwidth, and the pilot bandwidth can be expressed as a function of the unknown density *f*. The HSJM estimator estimates the unknown functionals of *f* with a kernel estimator that makes use of normal rule-of-thumb bandwidths. Since *f* is estimated, the pilot bandwidth can be estimated, which is then used to estimate the pilot density. The pilot density is then substituted into the unknown density *f*, that is required in the closed-form solution of the HSJM bandwidth.

More formally, the HSJM closed-form bandwidth solution is then given by

$$h = \left[\frac{R(K)}{\left(\int x^2 K(x)dx\right)^2 R(\hat{f}''_{g(h)})}\right]^{\frac{1}{5}} N^{-\frac{1}{5}}, \tag{5}$$

where $R(K)$ is given in Eq. 2, $K(\cdot)$ is the selected kernel function and $\hat{f}''_{g(h)}$ is the pilot kernel density estimate with bandwidth $g(h)$ given by

$$g(h) = C(K)\left[\frac{R(f'')}{R(f''')}\right]^{\frac{1}{7}} h^{\frac{5}{7}}, \tag{6}$$

where *f* is estimated with a kernel estimator with a normal rule-of-thumb bandwidth, *C(K)* is a constant determined by the kernel, $K(t)$, and is given by

$$C(K) = \left[\int t^2 K(t)dt\right]^{\frac{2}{5}} \left(\int K(t)^2 dt\right)^{\frac{4}{5}}. \tag{7}$$

We see in the formulation of Eq. 5 that the only variable to be solved is the bandwidth *h* and that *h* occurs on both sides of the equation since the pilot bandwidth is a function of *h*. To solve for the optimal value of *h*, denoted as h_{HSJM}, the objective function in Eq. 5 is initialised with a value for *h* and the right-hand side of the equation is calculated which will yield an updated value for *h*. The right-hand side of Eq. 5 is then solved again with the updated value of *h*, and the process is repeated iteratively until the bandwidth, *h*, converges to the optimal bandwidth h_{HSJM}.

B. Data

We compare the performance of the above mentioned conventional bandwidth estimators on real-world pattern recognition tasks, and distinguish between two types of real-world datasets, namely datasets with no independent test set, and datasets with separate train and test sets. We refer to the real-world datasets with no independent test sets as cross-validation (CV) datasets, and to the datasets with separate train and test sets simply as real-world (RW) datasets.

We select eight *CV datasets* (with no independent test sets) from the UCI Machine Learning Repository [8] for the purpose of simulation studies. These datasets are selected to cover a wide range of applications in pattern recognition, and to be representative of the samples sizes and dimensionalities of typical pattern recognition problems. The selected datasets are the Old Faithful, Balance-Scale, Iris, Diabetes, Heart (Statlog), Vehicle and Waveform datasets.

The *Old Faithful* dataset consists of 272 observations of eruptions of the Old Faithful geyser in the Yellowstone national park. The observations consist of 2 measures, namely the duration since the previous eruption and the duration the observed eruption. There are no class labels assigned to the observations.

The *Balance-Scale* dataset consists of 625 observations of a balance scale that is tipped to the left, balanced or tipped to the rights. The observations consist of 4 features that measure the left weight, distance to the left, right weight and distance to the right. The classification of each observation is the positioning of the scale, thus left, right or balanced.

The *Iris* dataset consists of 150 observations of Iris flowers. Each observation consists of 4 measures of an Iris flower, and the classification of each observation is the type of Iris species to which the flower belongs, namely Iris Setosa, Iris Versicolour or Iris Virginica.

The *Diabetes* dataset consists of records of 768 patient records obtained from the National Institute of Diabetes and Digestive and Kidney Diseases. Each observation consists of 8 features that may be used to predict the onset of diabetes. The classification of each observation is a diagnostic of whether the patient shows signs of diabetes according to the World Health Organization criteria.

The *Heart (Statlog)* dataset consists of 270 patient records that each consist of 13 features that may be used to predict the presence of heart disease. The classification of each observation is the absence or presence of heart disease in a patient.

The *Vehicle* dataset consists of 946 observations of vehicle silhouettes. The observations consist of 18 features that describe the scale independent features and heuristic measures of a vehicle silhouette. The classification of each observation is the type of vehicle model, namely a bus, Chevrolet van, Saab 9000 or Opel Mantra 400.

The *Waveform* dataset consists of 5000 observations from waveforms generated from 2 or 3 base wave forms with added noise of unit variance. Each observation is sampled from one of three waveforms and is described by 21 features. The classification of each observation is the type of waveform.

The *Ionosphere* dataset consists of 351 radar observations of the ionosphere measured at Goosebay, Labrador. The observations consist of 17 pairs of complex values resulting from the radar returns. There are thus 34 features per observation, and the classification of each observation is whether radar returns show evidence of structure in the ionosphere (measured by collisions with free electrons) or whether signals mostly passed through the ionosphere.

We summarise the most important dataset properties in Table I and denote the number of classes as "C", the dimensionality as "D" and the number of samples as "N".

TABLE I. CV DATASET SUMMARY

Dataset	**C**	**D**	**N**
Old Faithful	1	2	272
Balance-Scale	3	4	625
Iris	3	4	150
Diabetes	2	8	768
Heart (Statlog)	2	13	270
Vehicle	4	18	946
Waveform	3	21	5000
Ionosphere	2	33	351

We also select two *RW datasets* (with independent train and test sets) from the UCI Machine Learning Repository [8] for the purpose of simulation studies. These data sets are selected to have relatively high dimensionalities, since high-dimensions are typical of many real-world pattern recognition tasks. The selected datasets are the Segmentation, Landsat and Optdigits datasets.

The Statlog Image *Segmentation* dataset consists of 2100 train and 210 test observations, each corresponding to 3x3 pixel region randomly selected from 7 outdoor images. The observations consist of 18 features that describe various properties of the region, such as the position, contrast with neighbouring pixels, intensity and color properties of the region. The classification of each observation is the class of segment to which the center pixel of the 3x3 region belongs, compared to the segmentation class of hand segmentations. The segmentation classes are brickface, sky, foliage, cement, window, path and grass.

The *Landsat* satellite dataset consists of 4435 train and 2000 test observations. Digital satellite images of the same scene were taken at four different spectral bands, two in the visible region and two in the near-infrared region. Each observation in the dataset corresponds to measurements in the four spectral bands for a 3x3 pixel region, which results in 36 features per observation. The classification of each observation is the class of land coverage of the centre pixel in the 3x3 region. The land coverage classes are: red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble and very damp grey soil.

We summarise the most important properties of the RW datasets in Table II and denote the number of classes as "C", the dimensionality as "D", the number of training samples as "Ntr" and the number of test samples as "Nte".

TABLE II. RW DATASET SUMMARY

Dataset	C	D	Ntr	Nte
Segmentation	7	18	2100	210
Landsat	6	36	4435	2000

C. Data pre-processing

Principal Component Analysis (PCA) is performed on all CV and RW datasets as a pre-processing step prior to density estimation. PCA is used to reduce the dimensionality of datasets by performing a linear transformation that re-aligns the feature axes to the directions of most variation, and thus minimizes the variance orthogonal to the projection. Features with smallest eigenvalues (or variance in the transformed feature space) may thus be disregarded. PCA also ensures that the features of the transformed feature space are orthogonal, thus ensuring the feature are decorrelated.

The dimensionalities of all datasets are reduced with PCA in two ways. The first approach is to select only the transformed features that have eigenvalues equal or larger than 1% of the eigenvalue of the principle component; we denote this dimensionality reduction approach as PCA1. The second approach is to select only the 5 transformed features with largest eigenvalues. This approach allows us to compare estimators in this relatively low dimensional feature space, as opposed to the first approach where there will in some instances be more than 10 features selected; we denote this dimensionality reduction approach as PCA5.

We perform a separate class-specific transformation of each class within a dataset, since it has been shown [9] that this transformation is more effective in compressing features, than when all classes are transformed simultaneously.

D. Performance evaluation

CV datasets do not have an independent test set, and we therefore perform 10-fold CV for each class of a CV dataset to evaluate the performance. A density is estimated for each of the 10 folds, and the respective left out test folds are used to calculate the likelihood scores of the test data points. The likelihood test scores of all samples in the class are then combined and used to estimate the entropy of an estimator for the class.

RW datasets have independent test sets, and we therefore estimate for each class the density function of the training set, and calculate the likelihood scores of the samples belonging to the same class in the test set. The likelihood test scores are then used to estimate the entropy of an estimator for the class.

III. EXPERIMETNAL DESIGN

In this section we describe the methodology used to perform comparative simulation studies of the estimators and datasets described in Section II.

A. Experiment 1

In Experiment 1 we compare the performance of the Silverman rule-of-thumb, MSP, LSCV, ULCV, LLCV, HSJM and ML-Gauss estimators on the Old Faithful, Balance-Scale, Iris, Diabetes, Heart(Statlog), Vehicle, Waveform and Ionosphere datasets summarised in Table I. We make use of the estimator implementations in the Matlab KDE Toolbox [10].

The PCA5 and PCA1 class-specific transformations are performed on each class and the entropy score of each estimator is calculated per class for both transformations.

The LSCV, ULCV and LLCV estimators employ a golden section search to optimise the bandwidth for the respective objective functions, and the search is initialised with a bandwidth based on the average nearest neighbour distance. All implementations assume a Gaussian kernel, and the Silverman estimator is assumes a Gaussian reference distribution.

The LSCV, ULCV and LLCV scale the features independently to 1 standard deviation prior to bandwidth estimation, and the optimal bandwidth is re-scaled per dimension according to the initial standard deviations after estimation. We denote scaling with (s) at the end of the names of the scaled estimators. Since PCA is performed as a pre-processing step on all data, this is necessary to prevent the over smoothing of dimensions with small eigenvalues and the under smoothing of dimensions with large eigenvalues, when only a single bandwidth is estimated per dimension.

We also implement the ML Gauss estimator, which simply estimates the sample mean and covariance of each class. We make use of a full covariance matrix estimate for dimensionalities between 1 and 10, and make use of a diagonal covariance matrix for dimensionalities higher than 10.

B. Experiment 2

In Experiment 2 we compare the performance of the Silverman rule-of-thumb, MSP, LSCV, ULCV, LLCV, HSJM and ML-Gauss estimators on the Segmentation, Landsat, and Optdigits datasets summarised in Table II. Similar to Experiment 1, the PCA5 and PCA1 class-specific transformations are performed on each class and the entropy score of each estimator is calculated per class for both transformations.

The LSCV, ULCV and LLCV estimators employ the golden section search and scaling of features as described in Experiment 1.

IV. RESULTS

In this section we present the results of the simulation studies for Experiments 1 and 2 as described in Section III.

A. Experiment 1

Tables III and IV show the comparative CV entropy results of the conventional estimators on the CV datasets for class-specific PCA5 and PCA1 pre-processing respectively. Note that where the dimensionality of a dataset is less than 5, PCA5 will only decorrelate the data and the dimensionality thus remains the same. The dataset name is indicated with "DS", the class number with "C" and the dimensionality of the class after pre-processing with "K". We also abbreviate the dataset names with two letter acronyms and the ML Gauss estimator as "Gauss".

The *Old Faithful* dataset results in in Table III show that the ULCV, LLCV and HSJM estimators perform competitively on the 2-dimensional dataset, and that the ML Gauss estimator under performs.

The *Balance Scale* dataset results in Table III show that the Silverman, MSP and HSJM estimators perform competitively on all classes of the 4-dimensional dataset. Interestingly, the excellent performance of these estimators on the PCA5 dataset for class 2, does not hold for the PCA1 results in Table 2 - although they still perform competitively. We observe that for the PCA1 dataset, class 2 only has 3-dimensions. This implies that the fourth dimension (that has been dropped in PCA1), might have contained useful information for the purpose of density estimation.

The *Iris* dataset results in Tables III and IV show that the PCA5 and PCA1 transformations reduce the dataset to the same intrinsic dimensionality, the results are thus identical. We observe in these tables that the ML Gauss, MLE(init), Silverman and MSP estimators perform competitively on the 4-dimensional Iris dataset. The HSJM estimator consistently underperforms on all three classes.

The *Diabetes* dataset results in Table III show that the ULCV, LSCV, Silverman and MSP estimators perform competitively on the 5-dimensional dataset. The results in Table IV show that the performance of the Silverman and MSP estimators remains competitive on the 8 dimensional dataset and that the ULCV and LSC experience degradation in performance (relative to the other datasets). The LLCV, on the other hand, experiences a relative increase in performance as the dimensionality increases from 5 to 8. The HSJM and ML Gauss estimators underperform consistently on both the 5 and 8 dimensional datasets. We also observe an increase in entropy between the results for PCA5 and PCA1, which is due to the higher dimensionality of the PCA1 dataset.

The *Heart (Statlog)* dataset results in Table III show that the ULCV, LSCV, Silverman and MSP estimators, again, perform competitively on the 5-dimensional dataset. The results in Table IV show that the performance of the Silverman and MSP estimators remains competitive on the 13 dimensional dataset and that the ULCV and LSCV again experience a relative degradation in performance. The LLCV estimator underperforms on both datasets, and also experiences degradation in relative performance as dimensionality increases. The HSJM estimator, on the other hand, experiences a relative improvement in performance as dimensionality increases, and performs competitively on the 13 dimensional dataset. The relative performance of the ML Gauss estimator remains similar, and this estimator does not perform competitively on both the PCA5 and PCA1 datasets.

The *Vehicle* dataset results in Tables III and IV show that the Silverman and MSP estimators perform competitively on the 5-dimensional PCA5 dataset and on the higher-dimensional PCA1 dataset. The HSJM estimator experiences a relative increase in performance as dimensionality increases, and performs competitively on the PCA1 dataset. The ULCV, LSCV and ML Gauss estimators underperform on most classes of the PCA5 and PCA1 datasets.

The *Waveform* dataset results in Tables III and IV show that the ML Gauss estimator performs optimally across all classes on both the 5-dimensional PCA5 and 21-dimensional PCA1 datasets. The Silverman and MSP estimators perform competitively across all classes, and the HSJM estimator consistently underperforms on the PCA5 and PCA1 datasets. The ULCV, LLCV and LSCV estimators consistently underperform across all classes, and experience a drastic degradation in relative performance as the dimensionality increases from 5 to 21. Conversely, the relative performance of the ML Gauss estimator improves drastically as dimensionality increases.

The *Ionosphere* dataset results in Table III show that the ULCV, Silverman and MSP estimators perform competitively across all classes on the 5-dimensional PCA5 dataset. We see that the LSCV underperforms severely on class 1 of the PCA5 dataset, and that a valid bandwidth estimate was not obtained for the 33-dimensional class 1 of the PCA1 dataset in Table IV. We also note that the entropy scores of class 1 on the PCA1 dataset are much higher than the entropy scores of class 2 on the PCA dataset due to the large difference in dimensionality (33 as opposed to 12).

In summary, we have observed in this experiment that the Silverman and MSP estimators perform consistently well across all dimensions, and that the relative performances of ML-Gauss estimator improves with an increase in dimensionality. We also observed that the ULCV, LLCV and LSCV estimators generally did not perform well on dimensionalities of 9 and higher, and that the HSJM estimator significantly underperformed on the high-dimensional PCA1 Waveform and Ionosphere datasets.

B. Experiment 2

The Segmentation dataset results in Tables V and VI show that the Silverman and MSP estimators perform competitively on both the 5-dimensional PCA5 dataset, and the higher-dimensional PCA1 dataset. Again, the ULCV, LLCV and LSCV estimators experience a relative decrease in performance with an increase in dimensionality. The ML Gauss estimator underperforms on both the PCA5 and PCA1 datasets.

The Landsat dataset results in Tables VII and VIII show that the ULCV estimator performs optimally on the PCA5 dataset, and that the relative performance of the estimator degrades for the higher-dimensional PCA1 dataset. The LLCV and LSCV again experience a relative degradation in

TABLE III. CV ENTROPY RESULTS (PCA5)

DS	C	K	ULCV(s)	LLCV(s)	LSCV(s)	Silverman	MSP	HSJM	Gauss
OF	1	2	1.5114	1.5190	1.6933	1.6418	1.7041	1.5120	2.0179
BS	1	4	6.0706	6.4392	6.0706	5.4861	5.4534	5.4718	5.5393
	2	4	10.9974	10.9706	10.9974	-28.7634	-28.8503	-27.7191	4.2203
	3	4	6.0643	6.4294	6.0643	5.4947	5.4553	5.4409	5.5254
IR	1	4	5.7345	6.7486	5.7581	5.6587	5.5549	7.1666	5.6510
	2	4	4.9665	4.9535	4.9731	4.8840	4.7888	5.3478	4.6654
	3	4	4.9976	5.2579	5.1485	5.0947	5.0032	5.5725	4.9727
DB	1	5	7.2352	8.3182	7.2195	7.2378	7.2376	8.7124	7.6905
	2	5	7.4508	7.6156	7.4624	7.4759	7.4759	8.3736	7.6592
HS	1	5	8.1854	8.3171	8.1928	8.1676	8.1849	8.9343	8.2867
	2	5	8.2130	8.4184	8.2571	8.3159	8.2435	9.4678	8.2640
VC	1	5	8.5931	8.4109	8.5931	8.4350	8.5600	8.5136	8.9478
	2	5	8.7049	8.5827	8.7313	8.3850	8.4799	8.5244	8.7638
	3	5	7.7522	11.6123	7.8590	7.6507	7.6942	8.7665	9.5198
	4	5	8.2437	8.3692	8.2256	8.1073	8.2510	8.2330	9.4099
WF	1	5	8.2734	8.2803	8.4415	8.2850	8.2736	8.8006	8.2073
	2	5	8.2186	8.2510	8.3847	8.2345	8.2189	8.6721	8.1328
	3	5	8.1953	8.2139	8.3495	8.2147	8.1956	8.7014	8.1192
IS	1	5	9.7017	9.8104	20.9404	10.1500	9.8599	15.9452	9.8734
	2	5	8.9376	9.5393	9.1564	9.0250	9.1386	10.5080	10.7312

TABLE IV. CV ENTROPY RESULTS (PCA1)

DS	C	K	ULCV(s)	LLCV(s)	LSCV(s)	Silverman	MSP	HSJM	Gauss
OF	1	2	1.5114	1.5190	1.6933	1.6418	1.7041	1.5120	2.0179
BS	1	4	6.0706	6.4392	6.0706	5.4861	5.4534	5.4718	5.5393
	2	3	4.9890	5.1079	5.1893	4.8653	4.8488	4.8127	4.8764
	3	4	6.0643	6.4294	6.0643	5.4947	5.4553	5.4409	5.5254
IR	1	4	5.6716	6.7132	5.7023	5.6587	5.5549	7.1666	5.6510
	2	4	4.9665	4.9535	4.9731	4.8840	4.7888	5.3478	4.6654
	3	4	4.9976	5.2579	5.1485	5.0947	5.0032	5.5725	4.9727
DB	1	8	10.4729	9.9768	10.4729	10.0000	9.9474	11.3712	10.7987
	2	8	10.6695	10.5866	10.6695	10.5426	10.5469	11.1479	11.0914
HS	1	13	23.7675	24.1119	23.7675	18.6033	18.4294	18.3579	18.4897
	2	13	18.8418	21.4540	20.4495	17.7807	17.7338	17.9350	18.3348
VC	1	9	16.1516	15.4709	16.1516	10.9220	11.0842	10.9019	11.5029
	2	8	15.1835	14.5981	15.1835	10.2151	10.3520	10.2152	10.7001
	3	8	12.6065	20.9062	12.6065	9.3708	9.5106	10.1298	11.8372
	4	11	17.9895	14.8125	18.0032	12.1415	12.2843	12.2024	14.1832
WF	1	21	47.9183	46.4132	47.9183	27.1867	26.7686	38.3311	24.9595
	2	21	44.9225	46.4381	44.9225	29.6012	29.2137	38.9844	27.4697
	3	21	46.9280	49.5978	46.9280	29.5804	29.1566	40.6167	27.3635
IS	1	33	49.9981	50.0368	Inf	46.9883	44.0018	60.6990	52.1698
	2	12	17.6331	18.4547	17.6331	18.0641	16.6253	25.3506	17.1796

TABLE V. SEGMENTATION TEST ENTROPY RESULTS (PCA5)

C	K	NC	ULCV(s)	LLCV(s)	LSCV(s)	Silverman	MSP	HSJM	Gauss
1	5	330	8.6571	9.4309	8.6400	8.6097	8.5353	10.5575	9.1783
2	5	330	7.9876	7.7229	8.8720	7.7100	7.7971	10.6039	9.2186
3	5	330	6.6543	8.9703	Inf	7.0207	7.0412	13.8225	9.0371
4	5	330	8.7773	10.0652	14.5373	11.0827	10.0475	26.0267	10.0415
5	5	330	7.7285	16.3031	28.9613	8.0964	7.7062	25.3649	8.9870
6	5	330	8.1480	8.2896	7.9676	8.0393	8.0272	9.1536	8.6815
7	5	330	4.8831	5.0227	4.7681	5.6182	5.9123	4.8821	7.3298

TABLE VI. SEGMENTATION TEST ENTROPY RESULTS (PCA1)

C	K	NC	ULCV(s)	LLCV(s)	LSCV(s)	Silverman	MSP	HSJM	Gauss
1	13	330	19.7828	19.7140	19.7828	15.2068	15.0066	16.0416	19.4171
2	11	330	16.0282	19.8177	16.0282	12.9305	13.0771	15.7156	15.6992
3	12	330	10.1055	13.9903	Inf	10.5969	10.8379	20.6875	15.7573
4	11	330	13.8961	14.1103	13.8961	17.5817	15.9097	Inf	18.0036
5	11	330	14.8880	28.4377	16.5385	16.4508	14.8832	56.2277	17.1998
6	11	330	18.1200	16.0733	18.1200	11.3063	11.5558	11.2357	14.8150
7	10	330	9.8896	11.5070	9.8896	8.5376	8.8146	11.4790	11.1225

TABLE VI. LANDSAT TEST ENTROPY RESULTS (PCA5)

C	K	NC	ULCV(s)	LLCV(s)	LSCV(s)	Silverman	MSP	HSJM	Gauss
1	5	1533	8.4065	8.5952	8.4809	8.5251	8.5239	10.8315	9.1363
2	5	703	8.1881	8.1793	8.2670	8.5457	8.7830	9.8281	10.4275
3	5	1358	9.4269	11.3001	9.4937	9.5859	9.4692	12.6314	10.2360
4	5	626	8.6614	10.1450	8.7826	8.5052	8.5341	10.7932	9.9060
5	5	707	9.5970	9.5845	9.7385	9.6460	9.7217	10.9622	10.5202
6	5	1508	9.0323	12.6993	9.2426	9.2208	9.0951	14.0773	9.9167

TABLE VIII. LANDSAT TEST ENTROPY RESULTS (PCA1)

C	K	NC	ULCV(s)	LLCV(s)	LSCV(s)	Silverman	MSP	HSJM	Gauss
1	8	1533	11.3411	10.9108	11.3411	10.9399	10.8943	13.9897	11.8245
2	10	703	13.3700	12.4680	13.3700	12.3249	12.5838	14.8169	14.8997
3	14	1358	26.3911	38.3289	26.3911	17.1899	16.7990	16.8388	18.0135
4	13	626	21.6951	34.4233	21.6951	14.7206	14.7212	14.8199	16.8540
5	11	707	18.1305	20.7268	18.1305	14.7804	14.8053	15.6636	16.2852
6	11	1508	16.4696	28.7230	16.4696	15.1227	14.6888	17.6072	16.0888

performance as dimensionality increases. The Silverman and MSP estimators perform competitively on both the PCA5 and PCA1 datasets, and also experience a relative improvement in performance with an increase in dimensionality. The HSJM estimator underperforms on all classes on the PCA5 dataset, and experiences a relative improvement in performance in some classes for the PCA1 dataset. The ML Gauss estimator underperforms on both the PCA5 and PCA1 datasets.

In summary, we observed for CV and RW data that the ULCV estimator performed very competitively on the low-dimensional PCA5 datasets, but generally suffered severe performance degradation on the higher-dimensional PCA1 datasets. The ULCV, LLCV and LSCV also experienced severe performance degradation on high-dimensional PCA1 data. The performances of the Silverman and MSP estimators were very consistent across all dimensionalities and tasks, and were generally very competitive. The ML Gauss estimator experienced a relative performance improvement with increased dimensionality, and might be favoured in very high dimensional settings. The performance of the HSJM estimator is more difficult to predict, since it performs competitively on some PCA5 and PCA1 tasks, and on other tasks it underperforms.

V. CONCLUSION AND FUTURE WORK

We have shown in this comparative study that there are several regularities in the relative performance of conventional kernel bandwidth estimators across different tasks and dimensionalities. In particular, we found that the conventional ULCV, LLCV and LSCV estimators are not suitable for dimensionalities larger than 10, and that the Silverman and MSP estimators consistently performed competitively across all dimensions for the datasets investigated. We also noted that the HSJM estimator performance was more difficult to predict.

Given the reliable performance of the Silverman rule-of-thumb estimator, and the intuitive theoretical motivation that the Silverman estimator optimises the AMISE (assuming a Gaussian reference distribution), we conclude that the Silverman estimator is suitable for kernel bandwidth estimation and initialisation on pattern-recognition tasks, even in high-dimensional feature spaces. We therefore recommend that this estimator be used as the baseline initialisation for the iterative methods mentioned in the introduction, and will investigate the efficacy of this estimator for bandwidth initialisation in future work.

REFERENCES

[1] B.W. Silverman, "Density estimation for statistics and data analysis," Vol. 26. Chapman & Hall/CRC, 1986.

[2] M. Rosenblat, "Remarks on some nonparametric estimates of a density function," Ann. Math. Statits., 27, p. 832-837, 1956.

[3] D. Scott and S. Sain, "Multidimensional density estimation," Handbook of Statistics, vol. 24, pp. 229–261, 2005

[4] E. Barnard, "Maximum leave-one-out likelihood for kernel density estimation", in Proceedings of PRASA 2010.

[5] J. Leiva-Murillo and A. Atres-Rodriguez, "Algorithms for maximum-likelihood bandwidth selection in kernel density estimators," Pattern Recognition Letters, Vol. 33, p. 1717-1724, 2012.

[6] S.J. Sheather, "Density Estimation," Statistical Science, Vol. 19, No. 4, p. 588-597, 2004.

[7] G.R. Terrel and D.W. Scott, "Variable kernel density estimation," The Annals of Statistics, Vol. 20, No. 3, p. 1236-1265, 1992.

[8] C.L. Blake and C.J. Merz, "UCI repository of machine learning databases," 1998. [Online] Available: http:// www.ics.uci.edu/~mlearn/

[9] E. Barnard, "Visualizing data in high-dimensional spaces", in Proceedings of PRASA 2010.

[10] A. Ihler and M. Mandel, "Kernel density estimation toolbox for matlab," [Online] Available: http://www.ics.uci.edu/ ihler/code/