Research article Available online www.ijsrr.org ISSN: 2279–0543

International Journal of Scientific Research and Reviews

Density Estimation methods based on Mass

Alwin Pinakas J*

Department of Electronics and Computer Systems, KG College of Arts and Science, Coimbatore, IndiaEmail: japinakas@gmail.com

ABSTRACT

Density estimation is the ubiquitous base modelling mechanism employed for many tasks including clustering, classification, anomaly detection and information retrieval. Commonly used density estimation methods such as kernel density estimator and k-nearest neighbour density estimator have high time and space complexities which render them difficult to apply in problems with large data size with a moderate number of dimensions. This is the fundamental limitation in the algorithms.The density estimation method which stretches this limit to an extent in dealing with millions of data more easily and quickly. We analyse the error of the estimation using a bias-variance analysis. We then perform an empirical evaluation method by replacing density estimators with the density-based algorithms, namely, DBSCAN, LOF and Bayesian classifier, representing three different data mining tasks of clustering, anomaly detection and classification respectively. The results show that these estimation method significantly improves their time complexities, while maintaining or improving their task-specific performances in clustering, anomaly detection, and classification respectively.

KEYWORDS-density estimation; density-based algorithms;

***Corresponding author**

AlwinPinakas J

Department of Electronics and Computer Systems,

KG College of Arts and Science,

Coimbatore, India

Email: japinakas@gmail.com

INTRODUCTION

Density estimation is ubiquitously applied to various taskssuch as clustering, classification, anomaly detection andinformation retrieval. Despite its pervasive use, thereare no efficient density estimation methods thus far. Most existing methods such as kernel density estimator and k-nearest neighbor density estimator cannot be applied toproblems with even a moderate number of dimensions andlarge data size. This paper is motivated to study efficient method for density estimation. The threeexisting density-based algorithms, when employ the density estimator, set a new runtime benchmark that is ordersof magnitude faster.

1. The density estimation method studied has asignificant advantage over existing methods in termsof time and space complexities.
2. Establish the characteristics of the method through a bias-variance analysis.
3. Verify the generality of the method by replacingold density estimators with the threedensity-based algorithms.
4. Significantly simplify and speed up the current algorithms using set-based definitions instead of the common point-based definitions The density estimation method distinguishes itselffrom existing methods by:
5. Employing no distance measures in the density estimation process.
6. Having average case sublinear time complexity andconstant space complexity. Thus, it can be applied tovery large databases in which methods suchas kernel and k-NN density estimators are infeasible, the other density estimators are presented in Section II, In Section III we analyses the error producedby the new estimator by a bias-variance analysis and providea comparison of the estimation results between the density estimator in Section IV. A discussion of the related issues andthe conclusions are provided in the last two sections.

DENSITY ESTIMATION

The most commonly used, density estimation methods, namely kernel density estimator and k-nearest neighbor density estimator is discussed here.

1. Kernel Density Estimator

Let **x** be an instance in a *d*-dimensional space *Rd*. Thekernel density estimator (KDE) defined by a kernel function *K*(·) and bandwidth *b* is given as follows.

$$\bar{f}KDE(x) = \frac{1}{nb^d}\sum_{i=1}^{n} K\frac{(x-x_i}{b}$$

The difference **x** - **x***i* requires some form of distance measure; and *n* is the number of instances in the given dataset *D*.

2. K-Nn Density Estimator

A k-nearest neighbour (k-NN) density estimator can be expressed as follows

$$\bar{f}KNN(x) = \frac{|N(x,k)|}{n\sum_{x'\in n(x,k)} distance(x,x')}$$

Where$N(\mathbf{x}; k)$ is the set of k nearest neighbors to $\mathbf{x}$; andthe search for nearest neighbors is conducted over D of size n.Both KDE and k-NN density estimators have $O(n2)$time complexity and $O(n)$ space complexity in order to estimate the densities of n instances. Although there arevarious indexing schemes to speed up the search for nearest neighbor in order to aid the k-NN density estimator, theyare not satisfactory in terms of dealing with high dimensional problems and large data sets.

3. *Density Estimator Based On Mass*

Mass is more fundamental than density estimator can be constructed from mass. The key advantage of mass is that it can be computed very quickly. The density estimator based on mass inherits this advantage and executes significantly faster than density estimators such as KDE and k-NN.It raises the capability of density-based algorithms to handle large data sets to a new high level.A mass base function is defined as follows

$$m\big(T(x)\big) = \begin{cases} m & if\ x\ is\ in\ a\ region\ of\ T(.) \\ 0 & otherwise \end{cases}$$

where $T(\cdot)$ is function which subdivides the feature space into non-overlapping regions based on the given data set D; and m is the number of samples in a region of $T(\mathbf{x})$ in which $\mathbf{x}$ falls into.Ting and Wells shows that mass can also be effectively estimated using data subsets $Di \subset D$ ($i = 1,\ldots,t$) and its associated $Ti(\mathbf{x}|Di)$, where $|Di|= \varphi \ll n$ Each $Di$is sampled without replacement from D. The mass estimatedusing subsamples is defined as

$$\overline{mass}(\mathbf{x}) = \frac{1}{t}\sum_{i=1}^{t} \mathbf{m}(T_i(\mathbf{x}|\mathcal{D}_i)).$$

We now introduce the new density estimators based on mass (DEMass) and describe its implementation.Once mass is estimated, density can be estimated as a ratio of mass and volume.Thus, the density estimators based on mass functions m(T(x)) and $Ti(\mathbf{x}|Di)$ are defined respectively as

$$f_{\mathbf{m}}(\mathbf{x}) = \frac{\mathbf{m}(T(\mathbf{x}))}{nv}.$$

$$\bar{f}_{\mathbf{m}}(\mathbf{x}) = \frac{1}{t}\sum_{i=1}^{t} \frac{\mathbf{m}(T_i(\mathbf{x}|\mathcal{D}_i))}{\psi v_i}.$$

where$v$ and vi are the volumes of regions $T(\mathbf{x})$ and $Ti(\mathbf{x}jDi)$, respectively.We use the term DEMass to refer to density estimator.DEMasshas two key differences/advantages when compared to the one based on a kernel method or k-NN:f¯ **m**is estimated from t instances only which are significantly smaller than *D* in a large data set. It sums over *t* number of randomly generated regions; whereas $\bar{f}$*KDE*sums over *n* number of instances in *D*, and $\bar{f}$*kNN*also requires a search on the entire data set. For a large data set, $\bar{f}$ is prohibitively expensive to compute in these two methods.$\bar{f}$ **m**needs no distance measures.

ERROR ANALYSIS THROUGH BIAS-VARIANCE DECOMPOSITION

The density estimator based on mass (DEMass) $\bar{f}$ **m**(**x**) can be thought of as a random variable because of its dependence on *D* and its random subsamples *Di* (*i* = 1,…, *t*). Accordingly, we analyse Mean Squared Error (MSE) of¯*f***m**(**x**) from its true probability density *pd*(**x**). It is defined as

$$MSE(\bar{f}_{\mathbf{m}}(\mathbf{x})) = E\left[\{\bar{f}_{\mathbf{m}}(\mathbf{x}) - p_d(\mathbf{x})\}^2\right]$$

where the expectation *E*[·] is taken over the distribution of ¯*f***m**(**x**).The result indicates that the variance increases when level *h* increases. Also, the result does not change even if we use the higher order approximation because the term *pd*(**c***i*)=*vi* dominates in the above formula. The property of DEMass, revealed from this error analysis, is similar to that of the conventional kernel density estimator which shows a bias-variance trade off—the bias decreases as the kernel bandwidth *b* decreases but this increases the variance; and the reverse is true if the kernel bandwidth in increased [17]. The parameter *k* in k-NN density estimator has the same effect. In conclusion, DEMass has a comparable estimation of density with the kernel density estimator if both trade-off bias and variance equally well; and it is indeed the case in practice. Figure 1 shows the estimation result of a normal distribution using KDE and DEMass, respectively. It demonstrates that DEMass produces similar result to that generated by KDE, for different data sizes.Smoothing can be applied by increasing *b* for KDE or decreasing *h* for DEMass which produces the estimation results as shown in Figure 2.

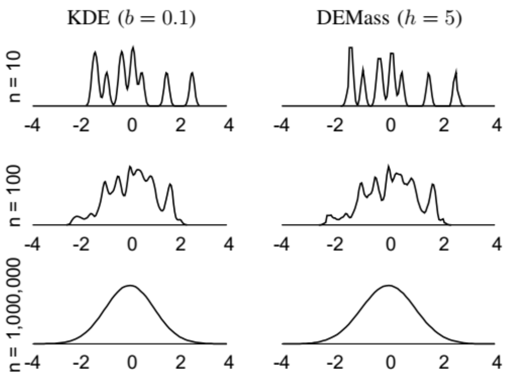


Figure 1: Example estimations of Kernel Density Estimator (with Gaussian kernel) using $b = 0.1$ and DEMass using $h = 5$ for different data sizes, $n = 10, 100, 1000000$. The true data distribution is a normal distribution.

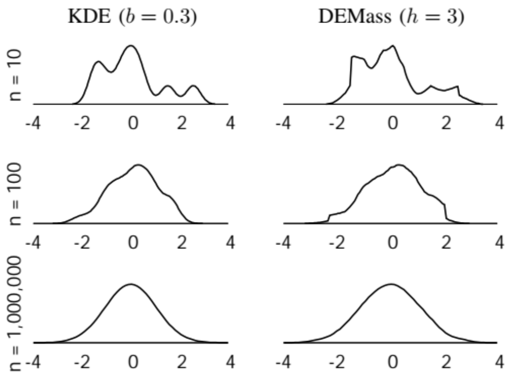
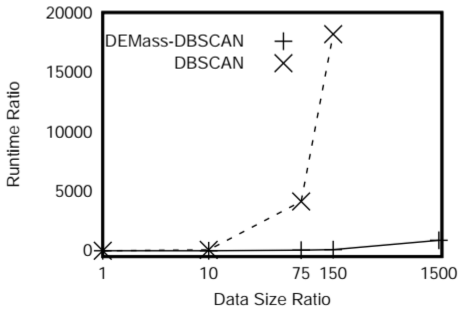


Figure 2: Example estimations of Kernel Density Estimator (with Gaussian kernel) using $b = 0.3$ and DEMass using $h = 3$ for the same data used in Figure 1.

The parameters used for DEMass are: *t* =1000 and = *n* when *n* =10;100; = 1000 when *n* =1000000.Note that in either settings shown in Figures1 and 2, the estimations of both KDE and DEMass approach the truedistribution as the number of instances increases.

COMPARISON RESULT

OneBig and Pendigits. The OneBig data set has 20 attributes, 9 clusters and a total of 68000 instances. Thebiggest cluster has 50011 instances, and each of the other eight clusters has approximately 1000 instances. In addition, there are 10000 noise instances randomly distributed in the feature space. The Pendigits data set has 16 attributes.



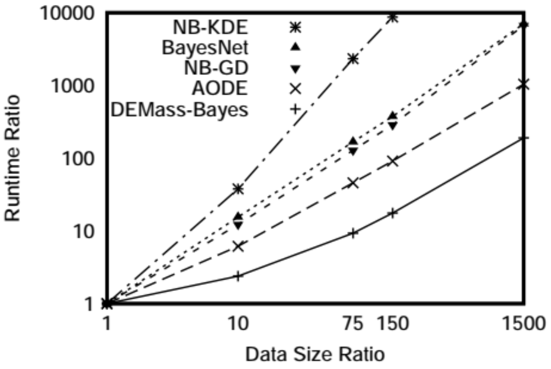
DEMass-DBSCAN vs DBSCAN in the 48-dimensional Ring-Curve-Wave-TriGaussian data set. Note that DBSCAN completed the task of the one-million data set (at data size ratio=150) in 36 days versus DEMassDBSCAN's 4.5 hours. Even with the 10-million data set, DEMass-DBSCAN completed it in 38 hours.Clustering results in the OneBig and Pendigits data sets for DEMass-DBSCAN (*h* = 3 for OneBig; *h* = 2for Pendigits) and DBSCAN (∈= 0*:*1 for OneBig; (∈= 0*:*2 for Pendigits).10 clusters. Each cluster has approximately 1100 instances which makes up a total of 10992 instances.The result in showed that DEMass-DBSCAN and DBSCAN for OneBig had the same clustering result in terms of F-measure and number of clusters; but DEMassDBSCAN ran faster than DBSCAN by a factor of 7. Note that DEMass-DBSCAN had correctly identified all but one of the 10000 noise instances; whereas DBSCAN correctly identified all of the noise instances.

	OneBig			Pendigits		
		DEMass-DBSCAN	DBSCAN		DEMass-DBSCAN	DBSCAN
Runtime		1145	8544		91	204
#cluster	[9]	9	9	[10]	47	65
#unassigned		10021	10005		2166	6251
F-measure		1.00	1.00		0.65	0.75

For Pendigits, the resultshowed that although DEMass-DBSCAN had a lower FMeasure than DBSCAN, it was better than DBSCAN inall other measures: it had only 20% instances unassignedwhereas DBSCAN had 57% instances unassigned; DEMassDBSCAN found 47 cluster whereas DBSCAN detected 65.Run time (in seconds) of a 10-fold cross validation for DEMass-Bayes and existing Bayesian classifiers: NB-KDE, NB-GD, BayesNet and AODE with default parameters.

Data set	DEMass -Bayes	NB-KDE	NB-GD	Bayes Net	AODE
Pendigits	2562	16	3	6	5
Magic04	423	93	4	9	5
Wave	237	24	2	4	3
RingCurve	227	22	3	4	4
LetRecog	4492	23	9	10	11
Shuttle	608	14	9	20	9
OneBig	3611	2361	28	101	32
MiniBooNE	3919	8594	134	375	119
Mulcross	1037	1832	29	64	20
CoverType	14088	1006	173	388	102

Scale up test: In order to examine how well the classifiers scale-up to large data size, we used the 48- dimensional Ring-Curve-Wave-TriGaussian data set, used in section VI-A. Data size was increased from 7000 to 70000, half-a-million, 1 million and 10 million. Figure showed the increase in runtime of DEMass-Bayes and the existing Bayesian classifiers. With the increase in data size by a factor of 10, 75, and 150, DEMass-Bayes increased its runtime by a factor of 2, 9, and 17. The closest contender AODE increased its runtime by a factor of 6, 45, and 91, followed by NB-GD (12, 128, 286), BayesNet (15, 167, 374), and NBKDE (38, 2345, 8721). Even with the data size increase by a factor of 1500, DEMass-Bayes only increased its runtime by a factor of 190, whereas BayesNet, NB-GD and AODE increased their runtime by factors of 7046, 6665 and 1038 respectively. DEMass-Bayes has a better scale up capability than the existing Bayesian classifiers.



CONCLUSIONS AND FUTURE WORK

The new density estimation method we introduced have two unique features which cannot be found in existingdensity estimation methods. First, it is the first density estimator that utilizes no distance measures. Second, it hasaverage case sublinear time complexity and constant space complexity. Existing density estimators must use a distance measure and have time and space complexities a lot worse than linear. The time and space complexities achieved a new benchmark for density-based algorithms, of what previously thought impossible. The bias-variance analysis reveals that the new density estimator has the same characteristic as kernel density estimator, i.e., both have a smoothing parameter used to trade-off between systematic error (bias) and random error (variance).Making full use of the features in the new density estimator, we show that two current algorithms, in the unsupervised learning setting from two key areas of data mining, can be significantly simplified through set-based definitions rather than the current point-based definitions. This has directly contributed to their improved time complexities. In the supervised learning setting, DEMass enables direct estimation of $p(\mathbf{x}jy)$ for the first time, without any assumption.Our evaluation shows that the new density estimator not only successfully replaces existing density estimatorsin three density-based algorithms, DBSCAN, LOF and Bayesian classifiers, but reduces their runtime to become algorithms with the lowest sub-linear time complexity. In addition, DEMass-DBSCAN, DEMass-LOF and DEMass-Bayes often achieve equivalent or better task-specific performances than DBSCAN, LOF and existing Bayesian classifiers. Our result implies that most, if not all, density-based algorithms can reap the immediate benefit of significantlylowering their time complexities by simply replacing the existing density estimators with the new one, with a potential further improvement in the task-specific performance. Future work has three directions. First, we will apply the density estimator in existing algorithms in moreareas. We will ascertain whether there are areas in which thenew density estimator cannot replace existing density estimators. Second, compare DEMass-density-based approacheswith mass-based approaches to determine their relativestrengths and weaknesses. Third, we will explore DEMass’sability to deal with high dimensional problem.

REFERENCES

1. Achtert, H.-P. Kriegel, and A. Zimek. Elki: A software system for evaluation of subspace clustering algorithms. In*Proceedings of the 20th International Conference on Scientificans Statistical Database Management*, 2008; 580–585.
2. A. Asuncion and D. Newman. UCI machine learning repository, 2007.
3. K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is ”nearest neighbor” meaningful? In *Proceedings of the 7th International Conference on Database Theory*, 1999; 217–235,
4. A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006; 97–104
5. M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proceedings of ACM SIGMOD*, 2000; 93–104.

6. P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 1997; 426–435.
7. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of KDD*, 1996;226–231.
8. A. Hinneburg, C. C. Aggarwal, and D. A. Keim. What is the nearest neighbor in high dimensional spaces? In *Proceedings of the 26th International Conference on Very Large Data Bases*, 2000; 506–515.
9. A. Hinneburg and D. A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proceedings of KDD*,. AAAI Press, 1998; 58–65
10. M. Kavitha Proportional learning on noise removal methods, In *Imperial Research of Interdisciplinary research. 2017*.