Low Complexity Algorithm for Probability Density Estimation Applied in Big Data Analysis

[1]Smail TIGANI, [2]Mouhamed OUZZIF, [3]Abderrahim HASBI and [4]Rachid SAADANE
[1]RITM/ESTC, National High School of Electricity and Mechanics. Casablanca, Morocco
[2]RITM/ESTC, High School of Technology. Casablanca, Morocco
[3]RSE/EMI, Mohamadia School of Engineering. Rabat, Morocco
[4]SIRC/LaGeS-EHTP, Hassania School of Labor Works. Casablanca, Morocco

ABSTRACT

Running inference algorithms on a huge quantity of data knows some perturbations and looses performance. One of Big Data aims is the design of fast inference algorithms able to extract hidden information on a big quantity of data. This paper proposes a new low complexity algorithm for probability density estimation given partial observations. In order to reduce the complexity of the algorithm, a finite numerical data support is adopted in this work and observations are classified by frequencies to reduce there number without loosing significance. By frequency classification we mean, the mapping from the space containing all observed values to a space containing each observable value associated with its observation frequency. This approach relies on Lagrange interpolation for approximating the frequencies with a polynomial function and then build the probability density function. To prove the reliability of the approach, a simulation is done and results shows the convergence of discussed parameters to the expected values. Big Data field can benefit considerably from proposed approach to achieve density estimation algorithms goal with low cost.

Keywords:

Probability Density Estimation, Big Data, Polynomial Interpolation, Classification, Algorithms and Complexity

1. INTRODUCTION

In real life, there is some measurable physical phenomenons needing monitoring and management permanently according to [1] wrote by V. S. Kumar Samparthi and al (2010), while the management of environments behaving randomly is harder than deterministic ones. Probability density estimation (PDE) is considered one of famous tools giving clear ideas about a random process behavior. According to H. Cheol Cho and al (2008) in [2], several parametric and non parametric probability density estimator exists and widely applied in many engineering fields such as artificial intelligence : machine learning, pattern recognition and in econometric...

This paper presents a new method for probability density estimation based on Lagrange interpolation given partial data. Proposed approach is implemented in algorithms and programmed with C++ programming language. The program includes a simulation part emulating the random environment by the generation of random observations and the estimation of the density... Computational complexity in time is also optimized than traditional approaches.

The content of his paper is organized as follows : the section 2 lists some related works and some critics and limits discussion, while section 3 introduces the mathematical models of the proposed approach eliminating limits discussed in section 2. Algorithms design and computational complexity analysis are reported to the section 4, while simulation results and discussions are presented in section 5. Finally, a conclusion with some perspectives in last section.

2. RELATED WORKS

A. Assenza and al (2008) summarizes, in [3], some probability density estimation methods and affirms that density estimation gives better results than traditional tools of data analysis like Principal Component Analysis. In the same way, Adriano Z. Zambom and al (2013) adds, in [4], that kernel density with smoothing is the most used approach. All those approaches, treats mathematical aspects but did not discussed implementation sides and computational complexity aspects. A. Sinha and al (2008) discussed in [5] algorithmic cost in time of those methods and optimized computational complexity of Kernel density estimator using clustring... Normally, the estimation of complexity in time must includes costs of all functions in global expression. Exponential function cost must be included in Kernel complexity estimation whatever its complexity class.

On the other hand, L. Kamberi and al (2011) introduces ,in [6], some application fields of interpolation functions. By interpolation we mean, the building of a deterministic function representing the data cloud (x_i, y_i) collected from an environment. In addition, it is so evident that the Lagrange interpolation is a quadratic complexity algorithm $\Theta(n^2)$ for each computed point.

Real added value of mathematical models is in its implementation in a concrete field. It would be more interesting if it is implemented economically with a low cost. The specificity this work, in the one hand, is the use of Lagrange interpolation as

a new way to build a probability density function. In the other hand, developed models are implemented on algorithms and computational complexity of all the process is optimized using finite state approach and classification by frequencies.

3. MATHEMATICAL MODELS BUILDING

3.1 Probability Density Support

Let $R_X = \{e_1, ..., e_p\}$ be all possible observations set fixed from the beginning of the process and $\varphi : R_X \to \mathbb{N}$ an application associating each observation with an integer representing the frequency of observation. By the frequency we mean, the number of observation of each value in R_X : *if the value e_3 is observed 10 times for example during all the process, then we have $\varphi(e_3) = 10$.*

Probability Density Support R_X is the numerical interval containing all observable values. Let $R_{X_{min}}$ be the smallest observable value and $R_{X_{max}}$ the biggest one. Formally, we write $R_{X_{min}} = \min(R_X)$ and $R_{X_{max}} = \max(R_X)$. It is evident so that each observable value is smaller, or equals, than the upper bound and greater, or equals, than the lower bound.

3.2 Occurrence Classification

Let S_0 be the set containing all observed values during a process of n observations. The goal if this subsection is the extraction of R_X from S_0. For instance, let suppose all observed values are $S_0 = \{0, 0, 0, 1, 0, 2, 1\}$, We have 0 repeated 3 times and 1 two times and 2 just one time... We must have finally $R_X = \{0, 1, 2\}$ and $\varphi(0) = 3, \varphi(1) = 2, \varphi(2) = 1$.

3.3 Lagrange Approximation

During a process of a big number of observations, that gives S_0 with m values. The algorithm extracts R_X and data $(e_i, \varphi(e_i))_{i=1...p}$. The use of Lagrange Theory concerning the approximation of data cloud with a polynomial function allows it to build a function ϕ passing from each value of φ. Formally :

$$\phi(x) = \sum_{i=1}^{p} \left(\varphi(e_i) \prod_{j \neq i} \frac{x - e_j}{e_i - e_j} \right) \tag{1}$$

3.4 Probability Density Function

The aim of this work is to estimate the probability density function given some historical observations using ϕ-function defined in equation 1. A function $\widehat{f}_X$ is a probability density function if it is positive for each value in the support in one hand, and the sum, for discreet process, on all the support is equals to 1. Formally, if $\widehat{f}_X(x) \geq 0, \forall x \in R_X$ and $\sum_{x_k \in R_X} \widehat{f}_X(x_k) = 1$. To assure the two conditions, let define $\widehat{f}_X$ as the following :

$$\widehat{f}_X(x) = \begin{cases} \frac{1}{m}.|\phi(x)| & , x \in R_X \\ 0 & , x \notin R_X \end{cases} \tag{2}$$

The condition of positivity is verified because each value is positive due to the use of the absolute value. The seconde condition is verified also due to the division by total observations m.

3.5 Main Parameters

Let $\widehat{f}_X$ be the density function obtained by the equation 2. This subsection focuses on the newt main parameters : the expectation denoted $E(X)$, the variance denoted $V(X)$ and F_X the distribution function.

The expectation represents the average of all the population, it is defined by the equation 3 :

$$E(X) = \sum_{x \in R_X} x \widehat{f}_X(x) \tag{3}$$

Moreover, the variance representing the homogeneity Classical definition of variance if given by the equation 4 :

$$V(X) = \sum_{x \in R_X} (x - \mu_X)^2 \widehat{f}_X(x) \tag{4}$$

The probability of observation of x_0 is given by $\pi(x_0) = \widehat{f}_X(x_0)$ and the classical definition of distribution function F_X is given by the probability of the event $(X \leq x)$. Formally :

$$F_X(x) = \sum_{x_k \leq x} \pi(x_k) \tag{5}$$

4. ALGORITHMS DESIGN

4.1 Observations Classifier Algorithm

This section focuses on algorithms implementing previous equations. The goal of the algorithm 1 is the extraction of the set R_X given S_0 discussed in subsection 3.2. See the algorithm :

Algorithm 1: Observations Classifier

Input: S_0 : All Observations
Output: R_X : Probability Support

```
R_X ← ∅
foreach i ∈ S_0 do
    Exists ← False
    foreach j ∈ R_X do
        if i = j then
            // Frequency of j
            φ(j) ← φ(j) + 1
            Exists ← True
    if Exists = False then
        φ(i) ← 1
        R_X ← R_X ∪ i
```

4.2 Complexity Analysis

Let $|R_X|$ be the cardinal of R_X : content number. The algorithm 1 classifies the observations by frequencies, that reduces the number from a very big number $|S_0| = m$ to a finite small one $|R_X|$. Computational complexity of this algorithm is $\Theta(m.|R_X|)$ with R_X is finite state with small cardinal. It becomes as consequence $|R_X|\Theta(m)$ witch is approximated with linear class $\Theta(m)$.

4.3 Lagrange Fitting Algorithm

Lagrange fitting given by equation 1 will be computed with the the algorithm 2 :

Algorithm 2: Lagrange Fitting

Input: x : Variable
Output: $\phi(x)$: Image of x

$(p, s) \leftarrow (1, 0)$
for $i = 1$ **to** $|R_X|$ **do**
  for $j = 1$ **to** $|R_X|$ **do**
    if $i \neq j$ **then**
      $p \leftarrow p * \frac{x - e_j}{e_i - e_j}$
  $s \leftarrow s + \varphi(e_i) * p$
return s

4.4 Complexity Analysis

Lines 1 and 2 of the algorithm 2 contains two affectations and line 6 contains 5 operations (one affectation and four arithmetic operations) and one test at line 5, all repeated $|R_X|^2$ times. Line 7 contains 3 elementary operations and the cost of φ which costs $|R_X|$ iterations. Last line containing the return instruction is considered also an operation.

The final cost in time of the algorithm 2 is $\Theta(7|R_X|^2+3|R_X|+3)$ witch is equals to $\Theta(1)$. Note that $|R_X|$ is finite whatever the observation number. **Normally, the computational complexity of Lagrange Polynomial Approximation is quadratic $\Theta(m^2)$, but the use of a finite state and frequency classification, allows to have a constant complexity $\Theta(1)$.**

4.5 Density Computer Algorithm

This subsection discussed the main algorithm using results of the last two algorithms 1 and 2. Probability density expression is given by equation 2, it will be computed with the the algorithm 3 :

Algorithm 3: Density Computer

Input: x : Variable
Output: $\widehat{f}_X(x)$: Image of x

if $(x \leq R_{X_{max}})$ *and* $(x \geq R_{X_{min}})$ **then**
  return $\frac{1}{m}|\phi(x)|$
return 0

4.6 Complexity Analysis

The line 1 is a test containing 3 partial tests, while line 2 contains a return instruction and 2 arithmetic operations. The computational complexity of the algorithm 3 includes the cost of algorithms 1 and 2 due to the call of ϕ-function. The absolute value function is estimated with 3 elementary operations at worst : a test, inversion and a return instruction. That gives totally 9 operations. Let suppose a big number m of data. Let $f_T(m)$ be the total cost in time of the main algorithm 3. We summarize final complexity, to compute one point of the density function, as : $9 + \Theta(m) + \Theta(1)$. Complexity class so is $f_T(m) = \max(9, \Theta(m), \Theta(1))$, that gives a linear complexity class $\Theta(m)$ compared with traditional method with quadratic complexity $\Theta(m^2)$.

5. SIMULATION AND DISCUSSION

This section focuses on the simulation done in order to prove the convergence of the method. It is developed with C^{++} and the program generate a MATLAB script containing all necessary data to plot curves. C^{++} programming technics are inspired from [7] of B. Eckel (2000) and [8] of H. Schildt.

5.1 Density and Distribution Simulation

Simulation program generates, in the first step, a uniform probability density function as a referential density which is equals to $\frac{1}{b-a}$ for each value in the interval $I = [a, b]$ and null out of it. Next step is generating three random uniform samples, on the same interval I, with different sizes : small, medium and big size sample. The agent making perception with the proposed approach analyses the three samples and estimates the density and the distribution function for each sample size in order to show the influence of the size on the function convergence.

The figure 1 shows how density and distribution function estimation converges to the referential one for each sample size :

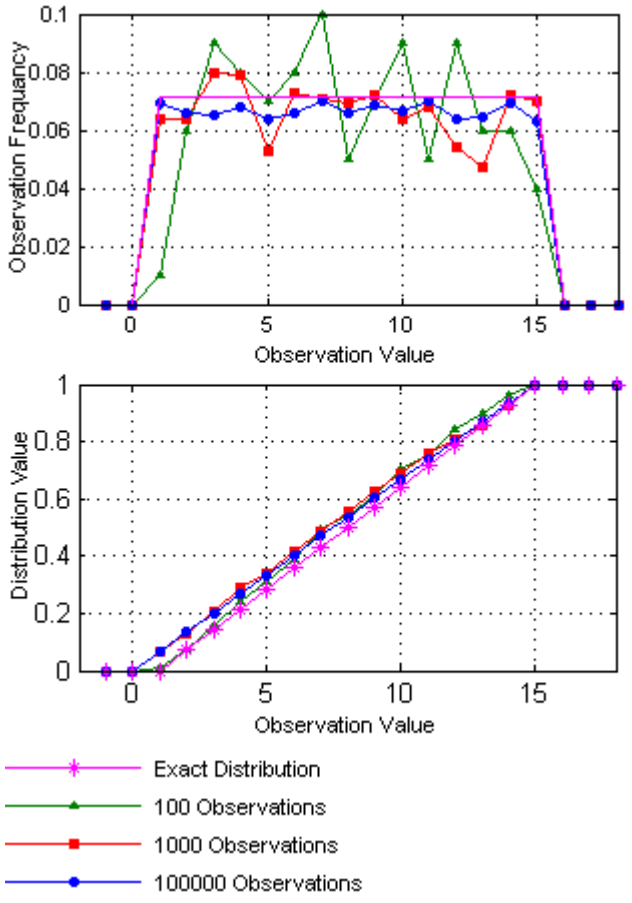

Fig. 1: Density and Distribution Function

5.2 Variance and Average Simulation

The aim of this section is the proof, by simulation, that the expectation and variance converges to referential ones. The principe of

this simulation is generating many uniform samples with graduated sizes and for each one the program computes the average and the variance with equations 3 and 4. The goal is to compare results with referential parameters for an uniform distribution given by $\mu = \frac{1}{b-a}$ and $\sigma^2 = \frac{(a+b)^2}{12}$ respectively. The residues between real and simulated values are defined as $\delta(\mu)$ and $\delta(\sigma)$ for the average and the variance respectively. The table 1 shows obtained results :

n	μ	$E(X)$	$\delta(\mu)$	σ^2	$V(X)$	$\delta(\sigma^2)$
1	7.00	5.33	1.66	14.08	69.81	55.72
2	8.00	8.05	0.05	16.33	14.69	1.63
3	8.00	6.95	1.05	16.33	19.96	3.63
4	8.00	6.66	1.33	16.33	17.36	1.02
5	8.00	8.10	0.10	16.33	16.07	0.25
6	8.00	7.28	0.71	16.33	17.20	0.87
7	8.00	8.16	0.16	16.33	17.59	1.26
8	8.00	7.45	0.55	16.33	17.88	1.55
9	8.00	7.53	0.46	16.33	19.03	2.70
10	8.00	6.96	1.03	16.33	15.29	1.03
11	8.00	7.52	0.47	16.33	17.46	1.13
12	8.00	7.50	0.49	16.33	15.57	0.76
13	8.00	7.17	0.82	16.33	18.18	1.85
14	8.00	7.41	0.58	16.33	18.15	1.82
15	8.00	7.49	0.50	16.33	19.36	3.02
16	8.00	8.05	0.05	16.33	17.50	1.17
17	8.00	7.19	0.80	16.33	17.53	1.19
18	8.00	6.85	1.14	16.33	19.13	2.80
19	8.00	6.93	1.06	16.33	17.13	0.79
20	8.00	7.65	0.34	16.33	14.24	2.09
21	8.00	7.97	0.02	16.33	16.96	0.63
22	8.00	7.25	0.75	16.33	17.94	1.61
23	8.00	7.56	0.43	16.33	17.49	1.15
24	8.00	7.51	0.48	16.33	16.98	0.65
25	8.00	7.38	0.61	16.33	18.88	2.55
26	8.00	7.57	0.42	16.33	16.23	0.10
27	8.00	7.70	0.29	16.33	17.49	1.16
28	8.00	6.84	1.15	16.33	17.49	1.15
29	8.00	7.92	0.07	16.33	18.19	1.86
30	8.00	7.30	0.70	16.33	14.16	2.16
31	8.00	7.00	0.99	16.33	15.44	0.88
32	8.00	7.45	0.54	16.33	18.99	2.66
33	8.00	7.90	0.09	16.33	17.80	1.46
34	8.00	8.19	0.19	16.33	14.09	2.23
35	8.00	7.73	0.26	16.33	18.98	2.65
36	8.00	7.92	0.07	16.33	16.84	0.51
37	8.00	7.28	0.71	16.33	16.78	0.45
38	8.00	7.38	0.61	16.33	19.21	2.88
39	8.00	8.06	0.06	16.33	15.30	1.02
40	8.00	7.30	0.69	16.33	18.61	2.28

Table 1. : Average and variance Simulation results

6. APPLICATION IN PREDICTION

M. Jakel (2013) describes in [9] predictor agent of a stock using genetic algorithms and time series, that can benefits from probability density estimation technics. Let define the next observation as the element in R_X having the highest probability. Let $e^\star$ be the next observation, it is given by :

$$e^\star = \arg \max_{e_k \in R_X} \pi(e_k) \tag{6}$$

The popular element $e^\star$ defined in the equation 6 is computed by the algorithm 4 :

Algorithm 4: Popular Element Finder

Input: R_X : Support
Output: $e^\star$: Popular Element

$pValue \leftarrow 0$
foreach $x \in R_X$ **do**
  if $pValue \leq \pi(x)$ **then**
    $e^\star \leftarrow x$
    $pValue \leftarrow \pi(x)$
return $e^\star$

L. Lebart and al (1995) talked, in [10], about classification methods and that allows us to find a link between classification and prediction capacity. The algorithm classifies all observable elements into two classes : popular class and non popular one. By popular class we mean, the subset frequently observed and this information allows to increase the visibility about the behavior of a chaotic environment. Let define a popular class $\mathcal{C}^*_{75}$ with 75% of appearance chance, formally :

$$\mathcal{C}^*_{75} = \{e \in R_X | \pi(e) \geq 0.75\pi(e^\star)\} \tag{7}$$

Based on the popular element $e^\star$ found by the algorithm 4, the algorithm 5 collects the popular class at 75% given by the equation 7. See the algorithm :

Algorithm 5: Popular Class Finder

Input: $e^\star$: Popular Element
Output: $\mathcal{C}^*_{75}$: Popular Class at 75%

$\mathcal{C}^*_{75} \leftarrow \emptyset$
foreach $x \in R_X$ **do**
  if $\pi(x) \geq 0.75\pi(e^\star)$ **then**
    $\mathcal{C}^*_{75} \leftarrow \mathcal{C}^*_{75} \cup x$
return $\mathcal{C}^*_{75}$

7. CONCLUSION

This work proposes a new method for probability density estimation, one of most important problems and widely applied in telecommunication traffic analysis, predictor agent and more... This work proposes, in addition to mathematical models, all necessary algorithms in order to implement the approach easily in specific domains. The specificity of the work is also the technics of classification used to optimize the computational complexity in time of algorithms witch adding to the new aspect an economical additional aspect.

Next work will focuses the implementation of the approach in order to estimate the interval of prediction with based on estimated density.

Acknowledgment

I would like to express all my gratitude to my supervisors Dr Mohamed OUZZIF, Dr Abderrahim HASBI and Dr Rachid SAADANE for excellent human behavior and technical support. Special thank to the Director of RITM Lab and the Director of High School of Technology - Casablanca, Dr Mounir RIFI.

Authors would like also to thank all reviewers for helpful comments and recommendations.

Thank to Dr Hafid GRIGUER, Mr Anis BOULAL, Miss Assya BENHIMA and Dr Hicham LAALAJ for all efforts done to encourage scientific research in EMSI Rabat.

Biography

Smail TIGANI : Is a network and telecommunication systems engineer and Phd Student in artificial intelligence and systems modelling. He worked as software engineer and now an information systems engineer and professor at Moroccan School of Engineering Science in Rabat. Recently, his researches focuses on the application of artificial intelligence on big data and performance analysis and optimization.

Mohammed OUZZIF : Is a Professor of Computer Science at High School of Technology of the Hassan II University. He has prepared his PHD at Mohammed V University in collaborative work field. His research interesting concerns distributed system and Formal description.

Abderrahim HASBI : Is a Professor of Computer Science at the Mohammadia School of Engineering of the University Mohamed 5 Agdal, Morocco. He is member of the Network and Intelligent systems Group and he has a lot of contributions researches.

Rachid SAADANE : He is currently an Associate Professor in the Electrical Engineering Department at Hassania School of Labor Works of Casablanca, Morocco. His research interests include array of UWB channel measurements modeling and characterization, mobile and wireless communications (GSM, WCDMA, TD/CDMA, LTE and LTE-A) and finally digital signal processing for wireless communications systems. Recently, he is intensively interested to the IR-UWB physical layer for WSN and WBAN. Rachid is an active reviewer of various international conferences and journals.

References

[1] V. S. Kumar Samparthi, Harsh K. Verma, *Outlier Detection of Data in Wireless Sensor Networks Using Kernel Density Estimation*, International Journal of Computer Applications, Vol. 5, No. 7, August 2010.

[2] H. Cheol Cho, M. Sami Fadali and K. Soon Lee, *Online Probability Density Estimation of Nonstationary Random Signal using Dynamic Bayesian Networks*, International Journal of Control, Automation and Systems, Vol. 6, No. 1, pp. 109-118, February 2008.

[3] A. Assenza, M. Valle, M. Verleysen, *A Comparative Study of Various Probabilty Density Estimaton Methods for Data Analysis*, International Journal of Computational Intelligence Systems, Vol. 1, No. 2, 2008, pp 188-201.

[4] Adriano Z. Zambom and R. Dias, *A Review of Kernel Density Estimation with Applications to Econometrics*, International Econometric Review, Vol. 5, No. 1, 2013, pp20-42.

[5] A. Sinha and S. Gupta, *Fast Estimation of Nonparametric Kernel Density Through PDDP, and its Application in Texture Synthesis*, International Academic Conference 2008 Visions of Computer Science, Vol. 5, No. 1, 2008, pp.225-236.

[6] L. Kamberi, T. Zenku, *Intrpolation of Functions with Application Software*, International Journal of Pure and Applied Mathematics, Vol. 73, No. 2, 2011, pp 219-225.

[7] B. Eckel, *Thinking in C++*, Vol. 1 Second Edition, January 13, 2000.

[8] H. Schildt, *C++: The Complete Reference*, Third Edition,.

[9] M. Jakel, *Genetically Evolved Agents for Stock Price Prediction*, International Journal of Inventive Engineering and Sciences, Vol. 10, No. 2, 2013, pp 21-35.

[10] L. Lebart, A. Morineay and M. Piron, *Statistique Exploratoire Multidimensionnelle*, ISBN 2 10 0028863, Dunod, Paris, 1995.