

LOCAL THRESHOLDING BERDASARKAN BENTUK UNTUK BINERISASI CITRA DOKUMEN

Aris Fanani, Putra Prima, M. Mahaputra Hidayat

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo, Surabaya

Email : fanani_aris@ymail.com, putraprima@gmail.com, riztra@gmail.com

ABSTRAK

Pada binerisasi dokumen, variasi intensitas lokal sering muncul dan merupakan akibat dari berbagai faktor seperti pencahayaan yang tidak merata, noise, dan tekstur dari kertas. Salah satu cara untuk mengatasi permasalahan tersebut adalah dengan metode local thresholding berdasarkan informasi bentuk. Akan tetapi proses komputasi yang lama diperlukan pada metode local thresholding citra dokumen yang memanfaatkan informasi bentuk. Hal ini disebabkan, setiap local window akan ditentukan nilai threshold-nya berdasarkan informasi bentuk. Paper ini mengusulkan penggabungan metode global thresholding dan local thresholding berdasarkan bentuk. Sehingga didapatkan kualitas thresholding yang lebih baik dengan proses komputasi yang lebih cepat. Pada local window yang memiliki intensitas rendah akan dilakukan proses thresholding secara global, sedangkan local window yang memiliki intensitas tinggi dilakukan local thresholding berdasarkan bentuk. Dari hasil uji coba didapatkan kualitas thresholding yang lebih baik dengan proses komputasi yang lebih cepat. Rata-rata keakuratan thresholding yang dicapai adalah 96,42%.

Kata Kunci: thresholding, local thresholding, thresholding berdasarkan bentuk.

1 PENDAHULUAN

Thresholding memainkan peranan penting dalam proses segmentasi citra dokumen. Tujuan dari thresholding ini adalah menemukan nilai threshold yang tepat untuk memisahkan objek dari background. Hasil thresholding merupakan citra biner, dimana semua piksel dengan tingkat keabuan yang lebih tinggi daripada nilai threshold akan diklasifikasikan sebagai objek dan sisa piksel diklasifikasikan sebagai background. Teknik thresholding dapat digunakan dalam berbagai aplikasi, seperti analisis citra biomedis dan identifikasi karakter tulisan tangan.

Pada binerisasi dokumen, thresholding merupakan pendekatan yang paling banyak digunakan. Sudah banyak pendekatan thresholding yang telah dikembangkan. Metode Otsu [1] merupakan salah satu contoh algoritma global thresholding, dimana metode ini memilih nilai threshold yang optimum dengan memaksimalkan between-class variance dari gray level pada bagian objek dan background. Metode lain adalah algoritma local thresholding. Dibandingkan dengan algoritma global thresholding, algoritma local thresholding lebih unggul dalam memilih nilai threshold berdasarkan pada variasi intensitas lokal.

Khusus untuk thresholding citra dokumen, variasi intensitas lokal sering muncul dan merupakan akibat dari berbagai faktor seperti pencahayaan yang tidak merata, noise, dan tekstur dari kertas. Algoritma local thresholding, bertujuan untuk menangani kesulitan yang disebabkan oleh intensitas

variasi, dimana nilai threshold ditentukan dari setiap piksel berdasarkan nilai grayscale-nya sendiri dan nilai grayscale tetangga. Beberapa algoritma local thresholding telah dikembangkan, diantaranya adalah metode Niblack [2]. Metode Niblack tidak efektif ketika background berisi tekstur berupa cahaya. Liu dan Srihari [3] mengusulkan algoritma untuk menghitung nilai threshold global dengan memanfaatkan informasi bentuk, sehingga diperoleh kualitas keluaran yang baik dengan memperhitungkan tekstur background dan intensitas cahaya. Metode ini mengukur lebar goresan dengan memanfaatkan run-length histogram. Terinspirasi dari metode tersebut, Jichuan Shi dkk [4] mengusulkan metode baru yang melakukan segmentasi semua karakter dari background dengan berbagai intensitas menggunakan seleksi nilai threshold lokal berdasarkan lebar goresan / karakter.

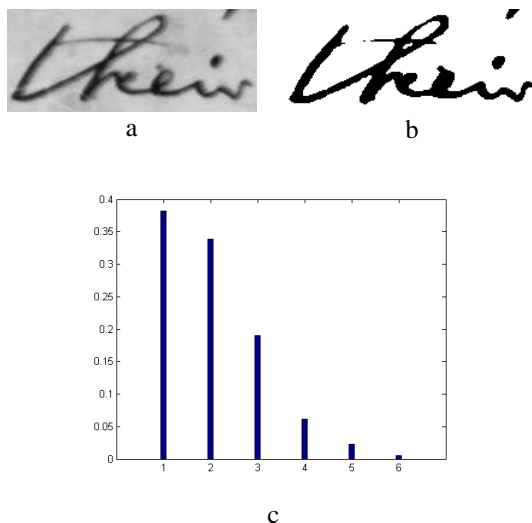
Meskipun menawarkan hasil yang memuaskan untuk citra dokumen, thresholding dengan pendekatan lokal yang diusulkan oleh Jichuan Shi dkk memiliki waktu komputasi yang tinggi. Hal ini disebabkan oleh proses penggunaan informasi bentuk untuk memperkirakan kandidat nilai threshold pada setiap local window. Pada paper ini diusulkan penggabungan algoritma global thresholding dan local thresholding berdasarkan bentuk untuk mengatasi permasalahan tersebut.

2 LOCAL THRESHOLDING BERDASARKAN BENTUK

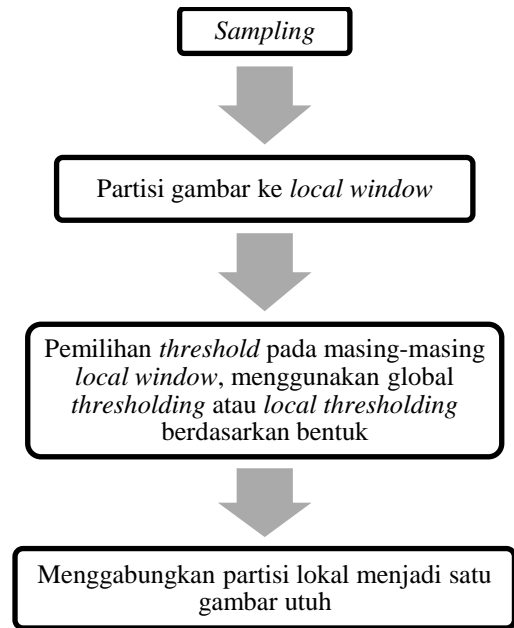
Metode *thresholding* yang diusulkan dalam paper ini dibagi menjadi empat proses, yaitu proses pengambilan *sample*, proses partisi/pembentukan *local window*, proses *local thresholding* dan proses penggabungan *local window*. Proses *sampling* (lihat Gambar 1) dilakukan oleh pengguna dengan memilih satu bagian terbaik dari citra yang berisi tulisan. Kemudian pada proses partisi, citra dokumen dibagi menjadi beberapa *local window*. Proses *local thresholding* dilakukan dengan cara mencari kandidat nilai *threshold* terbaik pada *local window*. Standar deviasi pada *local window* digunakan sebagai penentu metode *thresholding* yang akan digunakan, yaitu dengan *global thresholding* atau *local thresholding* berdasarkan bentuk. Proses terakhir adalah penggabungan masing-masing *local window* menjadi satu citra dokumen utuh yang telah dilakukan *thresholding*. Proses dari *local thresholding* berdasarkan bentuk tersebut, dapat dilihat pada Gambar 2.

2.1 Sampling

Distribusi lebar goresan karakter pada citra dokumen merupakan satu-satunya informasi yang digunakan dalam proses pengambilan sampel, dimana pengambilan sampel dilakukan dengan cara memotong sebagian kecil dari dokumen yang berisi karakter (Gambar 1(a)) dan memilih nilai *threshold* yang optimal dengan menggunakan metode Otsu (Gambar 1(b)). Selanjutnya, citra hasil *threshold* Otsu di hitung *distance transform*-nya dengan menggunakan metode Borgefors [5]. Setelah itu dibuat *distance histogram* $h_{sampling}$ (Gambar 1(c)) untuk potongan citra dokumen yang diambil.



Gambar 1. Proses Sampling



Gambar 2. Diagram Proses *Local Thresholding*

2.2 Pembentukan *Local Window*

Langkah selanjutnya adalah melakukan partisi untuk membentuk *local window*. Proses ini membagi citra dokumen menjadi ukuran 3x3, sehingga didapatkan 9 *local window* citra dokumen seperti pada Gambar 3. Citra dokumen dibagi menjadi ukuran 3x3 agar proses pemotongan lebih cepat. Selain itu juga menghasilkan *local window* yang merata untuk dapat dievaluasi variasi intensitasnya. Sehingga dapat ditentukan *local window* mana yang akan di-*threshold* menggunakan metode otsu dan *local window* mana yang akan di-*threshold* menggunakan *local thresholding* berdasarkan bentuk.

2.3 Thresholding pada *Local Window*

Proses selanjutnya dalam metode *local thresholding* berdasarkan bentuk adalah melakukan *thresholding* pada masing-masing *local window*. Proses ini membutuhkan inisialisasi nilai standar deviasi yang digunakan sebagai parameter penentuan metode *thresholding* yang akan digunakan.



Gambar 3. Hasil partisi *local window*

Langkah pertama yang dilakukan terhadap citra dokumen yang telah terbagi menjadi beberapa *local window* adalah menghitung nilai standar deviasi lokal seperti pada Gambar 4. Jika nilai standar deviasi lokal lebih rendah daripada standar deviasi global maka *local window* tersebut di-*threshold* menggunakan metode Otsu. Jika sebaliknya, proses *thresholding* dilakukan dengan metode *local thresholding* berdasarkan bentuk.

2.4 Thresholding berdasarkan Bentuk

Untuk setiap *local window*, sebuah kandidat nilai *threshold* dihasilkan oleh analisis sederhana dari intensitas histogram citra. Lembah yang dominan dipilih sebagai calon nilai *threshold* [3]. Untuk menghindari lembah yang tidak dominan akibat *noise*, maka perlu dilakukan proses *smoothing* yang diterapkan dalam analisis intensitas histogram. Metode estimasi *Probability Density Function* [4] diadopsi untuk mengatasi permasalahan tersebut. Metode ini secara otomatis menghasilkan distribusi kontinyu dari data histogram diskrit. *Probability Density Function* ditentukan dengan metode Silverman [6]:

$$H(x) = \frac{1}{nw} \sum_{i=1}^n K\left(\frac{x-x_i}{w}\right), \tag{1}$$

dimana X_i dalam kasus ini adalah nilai intensitas piksel dalam sebuah *local window* dari citra

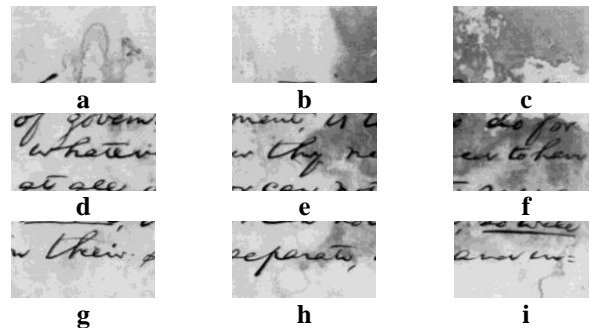
dokumen. n adalah jumlah pixel dalam *local window*, dan K adalah *kernel Gaussian*. w adalah hasil ukuran *smoothing* dari *local window* yang didapat dari persamaan (2), dimana σ adalah standar deviasi dan *IQR* merupakan nilai intensitas citra dokumen pada *local window*.

$$\begin{aligned} W &= 0.9A * n^{-1/5}, \\ A &= \min\left(\sigma, \frac{IQR}{1.34}\right). \end{aligned} \tag{2}$$

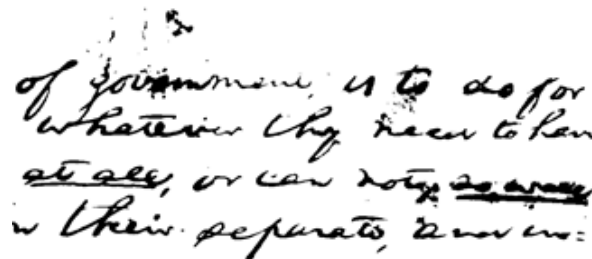
Setelah intensitas histogram dihasilkan, lembah (minimum) yang ditemukan akan dianggap sebagai kandidat nilai *threshold*. Untuk setiap kandidat nilai *threshold* $t \in T$, histogram pada *local window* dihitung kemiripannya dengan histogram *sampling*.

Bhattacharyya Coefficient (BC) [7] digunakan untuk mengukur kemiripan antara histogram *hsampling* dan histogram *test h(t)*. Semua histogram dinormalisasi. BC digunakan untuk mengevaluasi derajat kedekatan dari dua *probability density function*. Diantara semua kandidat nilai *threshold* yang dihasilkan dari satu *local window*, kandidat nilai *threshold* yang menghasilkan nilai BC yang maksimum akan dipilih sebagai nilai *threshold* pada *local window*, yaitu:

$$t^* = \arg \max [BC(h(t), h_{sampling})]. \tag{3}$$



Gambar 4. Penentuan metode *threshold* dengan standar deviasi global 0.50043, (a) standar deviasi 0.50129, (b) standar deviasi 0.50124, (c) standar deviasi 0.50129, (d) standar deviasi 0.48348, (e) standar deviasi 0.50129, (f) standar deviasi 0.50129, (g) standar deviasi 0.43487, (h) standar deviasi 0.50128, (i) standar deviasi 0.47613



Gambar 5. Hasil penggabungan *local window*

2.5 Penggabungan *Local Window*

Setelah mendapatkan nilai *local threshold* yang optimum untuk setiap *local window*, tahap terakhir dari metode yang diusulkan adalah menggabungkan *local window* tersebut untuk mendapatkan citra dokumen yang utuh. Seperti yang terlihat pada Gambar 5.

3 UJI COBA DAN PEMBAHASAN

Pada tahap uji coba, dataset yang digunakan berisi citra dokumen yang terdiri dari citra dokumen tulisan tangan dan citra dokumen tulisan cetak. Pada masing-masing citra dokumen terdapat sebuah *ground truth* dan citra dokumen terdegradasi, yang akan diuji dengan metode *thresholding* yang diusulkan.

Dalam pengujian digunakan lima citra dokumen seperti yang ditunjukkan pada Gambar 6. Untuk mengevaluasi performa dari algoritma yang diusulkan, metode yang diusulkan akan dibandingkan dengan metode Jichuan [4]. Uji coba perbandingan metode dilakukan untuk mengetahui seberapa unggul metode *local thresholding* berdasarkan bentuk yang diusulkan dibandingkan dengan metode yang telah dikembangkan Jichuan. Citra dokumen hasil *thresholding* metode yang diusulkan dapat dilihat pada Gambar 7, sedangkan citra dokumen hasil *thresholding* metode Jichuan dapat dilihat pada Gambar 8.

Kualitas secara kuantitas dari *thresholding* citra dokumen juga dievaluasi menggunakan dua metode yaitu, *misclassification error* (ME) dan *peak signal to noise ratio* (PSNR). ME didefinisikan sebagai hubungan antara citra dokumen hasil *thresholding* dengan *ground truth*. ME secara sederhana diformulasikan sebagai berikut:

$$ME = 1 - \frac{|B_0 \cap B_T| + |F_0 \cap F_T|}{|B_0| + |F_0|}, \quad (4)$$

dimana *background* dan *foreground* untuk *ground truth* dinotasikan B_0 dan F_0 , sedangkan B_T dan F_T merupakan *background* dan *foreground* hasil *thresholding* citra dokumen. Semakin kecil nilai ME menunjukkan bahwa hasil *thresholding* citra dokumen semakin baik.

Selain ME untuk mengukur kualitas hasil *thresholding* citra dokumen digunakan *Peak signal to noise ratio* (PSNR). Semakin tinggi nilai PSNR menunjukkan bahwa hasil *thresholding* citra dokumen semakin baik.

Untuk mengetahui kompleksitas komputasi dari metode yang diusulkan, dapat dilihat dari jumlah penggunaan *local thresholding* berdasarkan bentuk. Semakin sedikit penggunaan *local thresholding* berdasarkan bentuk, menunjukkan kompleksitas

komputasi metode semakin cepat. Hasil pengujian kedua metode tersebut dapat dilihat pada Tabel 1.

Hasil *thresholding* pada citra 1, 2 dan 5 dengan menggunakan metode yang diusulkan menghasilkan nilai ME relatif lebih besar dibandingkan dengan metode Jichuan. Hal ini terjadi karena karakteristik citra dokumen 1, 2 dan 5 memiliki jenis font yang sama. Pada metode yang diusulkan tidak semua *local window* di-*threshold* menggunakan *thresholding* berdasarkan bentuk. Sedangkan pada metode Jichuan semua *local window* di-*threshold* dengan *thresholding* berdasarkan bentuk.

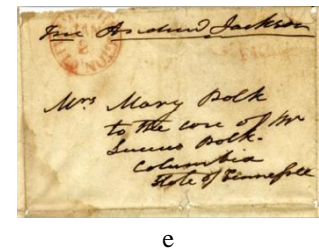
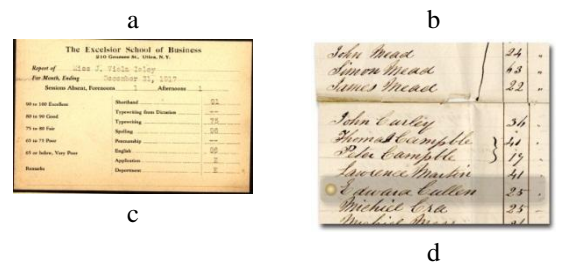
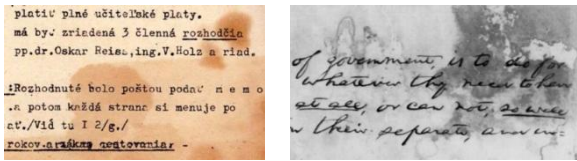
Pada citra 1, 2 dan 5, selisih ME antara metode yang diusulkan dengan metode Jichuan relatif kecil, yaitu sebesar 0.55, 1.01, dan 0.04. Akan tetapi jika dilihat dari kompleksitas komputasi antara metode Jichuan dan metode yang diusulkan terhadap citra 1 dan 2, metode yang diusulkan lebih rendah dalam kompleksitas komputasi. Sementara pada citra 5, kedua metode memiliki kompleksitas komputasi yang sama dengan selisih ME sebesar 0.04.

Hasil *thresholding* pada citra 3 dan 4, nilai ME metode yang diusulkan jauh lebih kecil dari metode Jichuan. Hal ini terjadi karena pada citra 3 dan 4 terdapat perbedaan jenis dan ketebalan *font*. Perbedaan jenis dan ketebalan *font* mengakibatkan performa *thresholding* berdasarkan bentuk menurun. Karena pada metode yang diusulkan tidak semua *local window* di-*threshold* berdasarkan bentuk maka hasilnya jauh lebih baik.

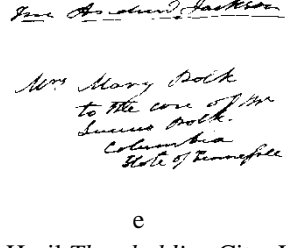
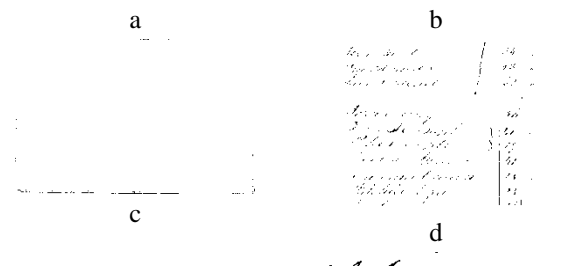
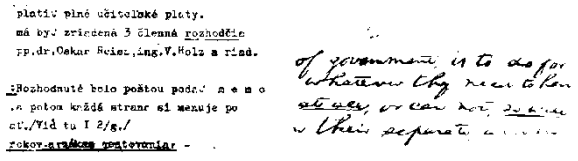
Tabel 1. Evaluasi Kinerja Metode yang diusulkan dan Metode Jichuan

Evaluasi	Citra	Jichuan	Metode diusulkan
PSNR	1	43.90	43.54
	2	50.68	42.01
	3	38.92	46.50
	4	32.87	48.68
	5	38.12	37.39
ME	1	1.18	1.73
	2	1.05	2.06
	3	8.00	3.38
	4	8.40	4.34
	5	6.35	6.39
Kompleksitas Komputasi	1	6	5
	2	9	6
	3	2	2
	4	6	2
	5	5	5

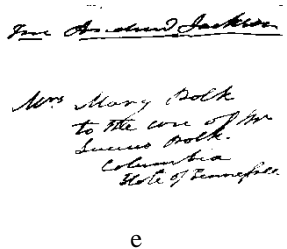
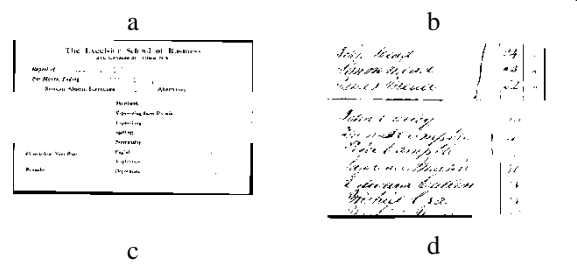
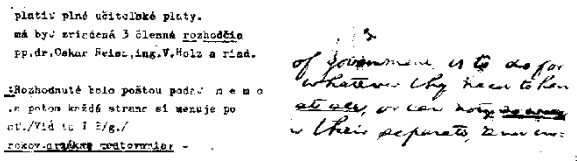
NB : Nilai dengan *font* yang ditebalkan menunjukkan hasil yang lebih baik



Gambar 6. Citra Dokumen Asli a. Citra 1, b. Citra 2, c. Citra 3, d. Citra 4, e. Citra 5



Gambar 8. Hasil *Thresholding* Citra Uji Metode Jichuan. Citra 1, b. Citra 2, c. Citra 3, d. Citra 4, e. Citra 5



Gambar 7. Hasil *Thresholding* Citra Uji Metode yang diusulkan. Citra 1, b. Citra 2, c. Citra 3, d. Citra 4, e. Citra 5

metode Jichuan. Dari uji coba yang dilakukan, menunjukkan bahwa hasil *thresholding* metode yang diusulkan memiliki kualitas yang hampir sama atau lebih baik dari metode Jichuan serta memiliki kompleksitas komputasi yang lebih rendah. Rata-rata nilai ME dari metode yang diusulkan sebesar 3,58%, dengan kata lain akurasi yang didapatkan sebesar 96,42%.

4 KESIMPULAN

Dari hasil uji coba dapat diambil kesimpulan sebagai berikut :

1. Metode yang diusulkan dapat melakukan proses *thresholding* lokal berdasarkan bentuk terhadap citra dokumen dengan baik, sehingga menghasilkan citra yang dapat dikenali secara visual.
2. Rata-rata akurasi dari metode yang diusulkan mencapai 96,42% dengan rata-rata jumlah *local window* yang di-*threshold* berdasarkan bentuk sebanyak 4. Sedangkan pada metode jichuan rata-rata jumlah *local window* yang menggunakan *local thresholding* berdasarkan bentuk sebanyak 5,6. Hal ini menunjukkan metode yang diusulkan memiliki waktu komputasi yang lebih rendah daripada metode sebelumnya.

Untuk kompleksitas komputasi pada citra 4 metode yang diusulkan lebih rendah sementara pada citra 3 dengan kompleksitas komputasi yang sama nilai ME metode yang diusulkan lebih baik sebesar 4,62. Dari Tabel 1 terlihat bahwa kompleksitas komputasi metode yang diusulkan lebih rendah dari

5 DAFTAR PUSTAKA

- [1] Otsu, N., 1979. "A Threshold Selection Method from Gray-level Histogram". IEEE Trans. Systems Man. Cybernet 9, 62–66.
- [2] Niblack, W. 1986. An Introduction to Digital Image Processing. Prentice Hall, Englewood Clifff, NJ.
- [3] Liu, Y., Srihari, S.N., 1997. "Document Image Binarization based on Texture Features". IEEE Trans. Pattern Anal. Machine Intell 19, 540–544.
- [4] Shi Jhicuan., 2011. "Shape Based Local Thresholding for Binarization of Document Image". Sciencedirect. Pattern Recognition Letters 33 : 24-32.
- [5] Borgefors, G., 1986. "Distance transformations in digital images". Comput. Vision Graph. Image Process 34, 344–371.
- [6] Silverman, B.W., 1986. Density Estimation for Statistics and Data Analysis. CRC Press, London, UK.
- [7] Bhattacharyya, A., 1943. "On a Measure of Divergence between Two Statistical Populations Defined by Their Probability Distributions". Bull. Calcutta Math. Soc 35, 99–109.