

Zastosowanie metody Cross Entropy Clustering w biometrii

2017.01.19

Krzysztof Misztal (krzysztof.misztal@uj.edu.pl)

Spis treści

1 Motywacja

2 Entropia

3 Ogólna idea “Cross Entropy Clustering”

4 Zastosowania

Spis treści

1 Motywacja

2 Entropia

3 Ogólna idea “Cross Entropy Clustering”

4 Zastosowania

Motywacja

- R skrypt 1
- R skrypt 2
- R skrypt 4

Spis treści

1 Motywacja

2 Entropia

3 Ogólna idea “Cross Entropy Clustering”

4 Zastosowania

Kompresja danych

Zbiór danych do zakodowania

$$x = \{x_1, x_2, \dots, x_n\}.$$

Zbiór symboli kodowych

$$C = \{c_1, c_2, \dots, c_n\}.$$

Uwaga

Zazwyczaj rozważamy $C = \{0, 1\}$.

Zbiór prawdopodobieństw

$$P = \{f_1, f_2, \dots, f_n\}.$$

Przykład

Przykład

ABCDBCDCDD

$$x = \{A, B, C, D\} \quad P = \{0.1, 0.2, 0.3, 0.4\}$$

Przykład

Przykład

ABCDBCDCDD

$$x = \{A, B, C, D\} \quad P = \{0.1, 0.2, 0.3, 0.4\}$$

1 A - 111 $I(A) = 3$

2 B - 000 $I(B) = 3$

3 C - 1 $I(C) = 1$

4 D - 0 $I(D) = 1$

Przykład

Przykład

ABCDBCDCDD

$$x = \{A, B, C, D\} \quad P = \{0.1, 0.2, 0.3, 0.4\}$$

1 A - 111 $I(A) = 3$

2 B - 000 $I(B) = 3$

3 C - 1 $I(C) = 1$

4 D - 0 $I(D) = 1$

1110
AD - ?? - CCCD

Przykład

Przykład

ABCDBCDCDD

$$x = \{A, B, C, D\} \quad P = \{0.1, 0.2, 0.3, 0.4\}$$

- 1 A – 111, $I(A) = 4$
- 2 B – 000, $I(B) = 4$
- 3 C – 1, $I(C) = 2$
- 4 D – 0, $I(D) = 2$

111,0 – 1,1,1,0
AD – CCCD

Jednoznacznie dekodowalny kod

Oczekiwana długość kodu to:

$$E(X) = \sum_{i=1}^n f_i \cdot l(x_i).$$

Nierówność Krafta-McMillana: warunkiem koniecznym, który musi spełniać kod, aby był jednoznacznie dekodowalny

$$\sum_{i=1}^n 2^{-l_i(x_i)} \leq 1.$$

Jednoznacznie dekodowalny kod

Oczekiwana długość kodu to:

$$E(X) = \sum_{i=1}^n f_i \cdot l(x_i).$$

Nierówność Krafta-McMillana: warunkiem koniecznym, który musi spełniać kod, aby był jednoznacznie dekodowalny

$$\sum_{i=1}^n 2^{-l_i(x_i)} \leq 1.$$

Przykład

$$(I) \ 2^{-3} + 2^{-3} + 2^{-1} + 2^{-1} = \frac{5}{4} > 1$$

$$(II) \ 2^{-4} + 2^{-4} + 2^{-2} + 2^{-2} = \frac{5}{8} < 1$$

Cel

Naszym celem jest minimalizowanie

$$L(l_1, \dots, l_n) := \sum p_i l_i$$

z warunkiem

$$\sum_{i=1, \dots, n} 2^{-l_i} \leq 1.$$

Cel

Naszym celem jest minimalizowanie

$$L(l_1, \dots, l_n) := \sum p_i l_i$$

z warunkiem

$$\sum_{i=1, \dots, n} 2^{-l_i} \leq 1.$$

Optymalne długości kodów zadane są

$$l_i = -\log_2 f_i.$$

Naszym celem jest minimalizowanie

$$L(l_1, \dots, l_n) := \sum p_i l_i$$

z warunkiem

$$\sum_{i=1, \dots, n} 2^{-l_i} \leq 1.$$

Optymalne długości kodów zadane są

$$l_i = -\log_2 f_i.$$

Zatem minimum jest osiągnięte dla kodów o długości zadanej przez entropię

$$h(X) := \sum_i f_i l_i = \sum_i f_i \cdot (-\log_2 f_i).$$

Kodowanie Huffmana

Przykład

ABCDBCDCDDF...

$$S = \{A, B, C, D, F\}$$

$$P = \{0.3, 0.3, 0.2, 0.15, 0.05\}$$

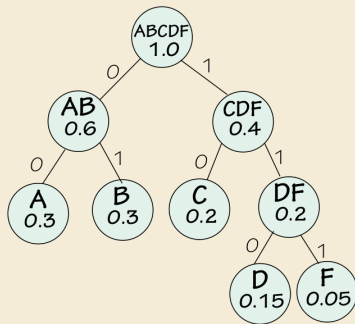
Kodowanie Huffmana

Przykład

ABCDBCDCDDF...

$S = \{A, B, C, D, F\}$

$P = \{0.3, 0.3, 0.2, 0.15, 0.05\}$



Kodowanie Huffmana

Przykład

ABCDBCDCDDF...

$S = \{A, B, C, D, F\}$

$P = \{0.3, 0.3, 0.2, 0.15, 0.05\}$

1 A – 00

2 B – 01

3 C – 10

4 D – 110

5 F – 111

Kodowanie Huffmana

Przykład

ABCDBCDCDDF...

$S = \{A, B, C, D, F\}$

$P = \{0.3, 0.3, 0.2, 0.15, 0.05\}$

1 A – 00

2 B – 01

3 C – 10

4 D – 110

5 F – 111

Przykład

$$\mathbb{E}(X) = 2.2$$

$$h(X) = 1.47863$$

Entropia krzyżowa (cross-entropy)

Założmy teraz, że chcemy zakodować zmienną losową Y (która przyjmuje takie same wartości jak zmienna X , czyli $x_1, \dots, x_n$, ale zgodnie z innym rozkładem gęstości g_i) kodem zdefiniowanym dla zmiennej losowej X (o rozkładzie f_i). Wtedy entropia krzyżowa (cross-entropy) $h^\times(Y\|X)$ opisująca oczekiwaną długość kodu dana jest przez

$$h^\times(Y\|X) := \sum_i g_i \cdot (-\log_2 f_i).$$

Entropia krzyżowa (cross-entropy)

Założmy teraz, że chcemy zakodować zmienną losową Y (która przyjmuje takie same wartości jak zmienna X , czyli $x_1, \dots, x_n$, ale zgodnie z innym rozkładem gęstości g_i) kodem zdefiniowanym dla zmiennej losowej X (o rozkładzie f_i). Wtedy entropia krzyżowa (cross-entropy) $h^\times(Y\|X)$ opisująca oczekiwaną długość kodu dana jest przez

$$h^\times(Y\|X) := \sum_i g_i \cdot (-\log_2 f_i).$$

Entropia krzyżowa jest minimalizowana, gdy $X=Y$, wtedy uzyskujemy po prostu entropię. Zatem entropię krzyżową możemy traktować jako sposób porównywania modeli, im bliżej nam do entropii tym lepszym przybliżeniem Y jest zmienna losowa X .

Entropia krzyżowa

$$h^{\times}(Y\|X) := \sum_i g_i \cdot (-\log_2 f_i).$$

Przykład

$$\begin{aligned} \mathbf{x} &= \{A, B, C, D\} & P &= \{0.1, 0.2, 0.3, 0.4\} \\ G &= \{0.25, 0.25, 0.25, 0.25\} & L &= \{0.7, 0.1, 0.1, 0.1\} \end{aligned}$$

Entropia krzyżowa

$$h^\times(Y\|X) := \sum_i g_i \cdot (-\log_2 f_i).$$

Przykład

$$\begin{aligned} X &= \{A, B, C, D\} & P &= \{0.1, 0.2, 0.3, 0.4\} \\ G &= \{0.25, 0.25, 0.25, 0.25\} & L &= \{0.7, 0.1, 0.1, 0.1\} \\ h(X) &= h^\times(Y\|X_P) = 1.8464 \end{aligned}$$

Entropia krzyżowa

$$h^\times(Y\|X) := \sum_i g_i \cdot (-\log_2 f_i).$$

Przykład

$$\begin{aligned} X &= \{A, B, C, D\} & P &= \{0.1, 0.2, 0.3, 0.4\} \\ G &= \{0.25, 0.25, 0.25, 0.25\} & L &= \{0.7, 0.1, 0.1, 0.1\} \\ h(X) &= h^\times(Y\|X_P) = 1.8464 \\ h^\times(Y\|X_G) &= 2 \end{aligned}$$

Entropia krzyżowa

$$h^\times(Y\|X) := \sum_i g_i \cdot (-\log_2 f_i).$$

Przykład

$$\begin{aligned} X &= \{A, B, C, D\} & P &= \{0.1, 0.2, 0.3, 0.4\} \\ G &= \{0.25, 0.25, 0.25, 0.25\} & L &= \{0.7, 0.1, 0.1, 0.1\} \\ h(X) &= h^\times(Y\|X_P) = 1.8464 \\ h^\times(Y\|X_G) &= 2 \\ h^\times(Y\|X_L) &= 3.04119 \end{aligned}$$

Entropia krzyżowa

$$h^\times(Y\|X) := \sum_i g_i \cdot (-\log_2 f_i).$$

Przykład

$$\begin{aligned} X &= \{A, B, C, D\} & P &= \{0.1, 0.2, 0.3, 0.4\} \\ G &= \{0.25, 0.25, 0.25, 0.25\} & L &= \{0.7, 0.1, 0.1, 0.1\} \\ h(X) &= h^\times(Y\|X_P) = 1.8464 \\ h^\times(Y\|X_G) &= 2 \\ h^\times(Y\|X_L) &= 3.04119 \end{aligned}$$

Entropia różniczkowa (differential entropy)

Entropia różniczkowa (differential entropy) zadana jest wzorem

$$H(X) := \int f(x) \cdot (-\log_2 f(x)) dx,$$

gdzie f oznacza rozkład prawdopodobieństwa zmiennej losowej X .

Entropia krzyżowa

Jeśli chcemy kodować zmienną Y (która opisana jest ciągłym rozkładem g) kodem optymalnym dla zmiennej X (opisanej ciągłym rozkładem f) to entropia krzyżowa

$$H^{\times}(Y\|X) := \int g(y) \cdot (-\ln f(y)) dy.$$

Entropia krzyżowa

Jeśli chcemy kodować zmienną Y (która opisana jest ciągłym rozkładem g) kodem optymalnym dla zmiennej X (opisanej ciągłym rozkładem f) to entropia krzyżowa

$$H^{\times}(Y\|X) := \int g(y) \cdot (-\ln f(y)) dy.$$

Przez entropię krzyżową zbioru X względem gęstości f rozumiemy

$$H^{\times}(X\|f) = -\frac{1}{|X|} \sum_{x \in X} \ln(f(x)).$$

Spis treści

1 Motywacja

2 Entropia

3 Ogólna idea “Cross Entropy Clustering”

4 Zastosowania

Algorytm EM (expectation–maximization)

W ogólnym przypadku algorytm EM znajduje

$$p_1, \dots, p_k \geq 0 : \sum_{i=1}^k p_i = 1, \quad (1)$$

oraz $f_1, \dots, f_k \in \mathcal{F}$, dla ustalonej rodziny $\mathcal{F}$, (zazwyczaj rodzina rozkładów normalnych) tworzących wypukłe kombinacje gęstości postaci

$$f := p_1 f_1 + \dots p_k f_k \quad (2)$$

najlepiej opisujących rozważany zbiór danych $X = \{x_1, \dots, x_n\}$.

Algorytm EM (expectation–maximization)

Optymalizacja jest rozważana względem funkcji kosztu zadanej przez MLE¹ (maximum likelihood estimation – funkcja największej wiarygodności)

$$\text{EM}(f, X) = -\frac{1}{|X|} \sum_{j=1}^n \ln (p_1 f_1(x_j) + \dots + p_k f_k(x_j)), \quad (3)$$

gdzie $|X|$ oznacza licznosc zbioru X .

¹Ponieważ w grupowaniu zazwyczaj minimalizujemy koszt, więc w takich przypadkach funkcja (3) rozważana jest z przeciwnym znakiem.

Idea algorytmu CEC

Celem algorytmu CEC jest minimalizowanie funkcji kosztu (podobnej do tej z równania (3), gdzie wymieniamy sumę na maksimum):

$$\text{CEC}(f, X) := -\frac{1}{|X|} \sum_{j=1}^n \ln (\max(p_1 f_1(x_j), \dots, p_k f_k(x_j))), \quad (4)$$

gdzie p_i dla $i = 1, \dots, k$ spełniają warunek

$$p_1, \dots, p_k \geq 0 : \sum_{i=1}^k p_i = 1.$$

■ R script 3

Idea algorytmu CEC

W przypadku podziału $X \subset \mathbb{R}^N$ na parami rozłączne zbiory $X_1, \dots, X_k$ takie, że każdy z nich jest “kodowany” z wykorzystaniem gęstości f_i możemy pokazać, że średnia długość kodu dla losowo wybranego elementu $x \in X$ wynosi

$$\text{CEC}(X_1, f_1; \dots; X_k, f_k) = \sum_{i=1}^k p_i \cdot (-\ln(p_i) + H^\times(X_i \| f_i)), \quad (5)$$

gdzie $p_i = \frac{|X_i|}{|X|}$.

Idea algorytmu CEC

Efektywne wyznaczanie mieszaniny modeli w CEC-u wymaga wyznaczenia optymalnej wartości kodowania zmiennej X względem rodziny gęstości $\mathcal{F}$, czyli

$$H^\times(X\|\mathcal{F}) := \inf_{f \in \mathcal{F}} H^\times(X\|f).$$

Idea algorytmu CEC

Warunek optymalności.

Podsumowując, dla zadanych rodzin gęstości $\mathcal{F}_1, \dots, \mathcal{F}_n$ celem algorytmu CEC jest podział zbioru X na k (zbiory mogą być puste) parami rozłączne zbiory $X_1, \dots, X_k$ takie, że wartość

$$\text{CEC}(X_1, \mathcal{F}_1; \dots; X_k, \mathcal{F}_k) := \sum_{i=1}^k p_i \cdot (-\ln(p_i) + H^\times(X_i \| \mathcal{F}_i)), \quad (6)$$

jest minimalna (gdzie $p_i = \frac{|X_i|}{|X|}$).

Grupowanie CEC – problemy

Generalnie są dwa problemy:

- jak policzyć

$$H^{\times}(X_i \| \mathcal{F}_i)$$

dla wybranej rodziny rozkładów $\mathcal{F}_i$,

Grupowanie CEC – problemy

Generalnie są dwa problemy:

- jak policzyć

$$H^{\times}(X_i \| \mathcal{F}_i)$$

dla wybranej rodziny rozkładów $\mathcal{F}_i$,

- jak dobrać rodzinę $\mathcal{F}_i$ tak, aby “spełniała” nasze oczekiwania.

Twierdzenie ([Tabor and Spurek, 2014])

Niech $X \subset \mathbb{R}^N$ będzie zbiorem z ustaloną średnią m_X oraz macierzą kowariancji Σ_X .

Wtedy

$$H^\times(X \| \mathcal{G}_\Sigma) = \frac{N}{2} \ln(2\pi) + \frac{1}{2} \text{tr}(\Sigma^{-1} \Sigma_X) + \frac{1}{2} \ln \det(\Sigma), \quad (7)$$

gdzie $\mathcal{G}_\Sigma$ oznacza wszystkie rozkłady normalne z macierzą kowariancji Σ .

Dowód tego twierdzenia znajduje się w artykule [Tabor and Spurek, 2014].

Klastrowanie CEC – dobór modelu

1. $\mathcal{G}_\Sigma$ – rozkłady normalne z ustaloną kowariancją Σ .

W tym przypadku będziemy mieli tendencję do podziału zbioru na grupy kul zadanych przez odległość Mahalanobisa $\|\cdot\|_\Sigma$ [Tabor and Spurek, 2014].



$$\Sigma_{\mathcal{G}_\Sigma}(X) = \Sigma$$

$$H^\times(y|\mathcal{G}_\Sigma) = \frac{N}{2} \ln(2\pi) + \frac{1}{2} \text{tr}(\Sigma^{-1} \Sigma_X) + \frac{1}{2} \ln \det(\Sigma)$$

2. $\mathcal{G}_{rI}$ – podrodzina $\mathcal{G}_{\Sigma}$, dla $\Sigma = rI$ oraz ustalonego $r > 0$.

Grupowanie będzie zadane przez sferyczne rozkłady normalne zadane przez macierz kowariancji rI , czyli kule o określonym promieniu proporcjonalnym do $\sqrt{r}$ [Tabor and Spurek, 2014].



$$\Sigma_{\mathcal{G}_{rI}}(X) = rI$$

$$H^{\times}(X|\mathcal{G}_{rI}) = \frac{N}{2} \ln(2\pi) + \frac{N}{2} \ln(r) + \frac{1}{2r} \text{tr}(\Sigma_X)$$

3. $\mathcal{G}_{(\cdot, I)}$ – sferyczne rozkłady normalne.

Grupowanie będzie zrealizowane przez sferyczne rozkłady normalne, czyli kule o dowolnym promieniu [Tabor and Spurek, 2014].



$$\Sigma_{\mathcal{G}_{(\cdot, I)}}(X) = \frac{\text{tr}(\Sigma_X)}{N} \mathbf{I}$$

$$H^\times(X \| \mathcal{G}_{(\cdot, I)}) = \frac{N}{2} \ln(2\pi e/N) + \frac{N}{2} \ln(\text{tr} \Sigma_X)$$

4. $\mathcal{G}_{\text{diag}}$ – rozkłady normalne o diagonalnej macierzy kowariancji.

Grupowanie zrealizowane będzie przez elipsy o osiach równoległych do osi układu współrzędnych [Tabor and Spurek, 2014].

$$\frac{\overline{\overline{\overline{|||}}}}{\overline{\overline{-}}}$$

$$\Sigma_{\mathcal{G}_{\text{diag}}}(X) = \text{diag}(\Sigma_X)$$

$$H^{\times}(X|\mathcal{G}_{\text{diag}}) = \frac{N}{2} \ln(2\pi e) + \frac{1}{2} \ln(\det(\text{diag}(\Sigma_X)))$$

5. $\mathcal{G}_{\lambda_1, \dots, \lambda_N}$ – rozkłady normalne o macierzy kowariancji z zadanymi wartościami własnymi $\lambda_1, \dots, \lambda_N$ takimi, że $\lambda_1 \leq \dots \leq \lambda_N$.

Grupowanie będzie dzieliło zbiór na elipsy o ustalonym kształcie, ale obrócone o dowolny kąt [Tabor and Misztal, 2013].


$$\Sigma_{\mathcal{G}}(X) = \Sigma_{\lambda_1, \dots, \lambda_N}$$
$$H^{\times}(X \| \mathcal{G}_{\lambda_1, \dots, \lambda_N}) = \frac{N}{2} \ln(2\pi) + \frac{1}{2} \sum_{i=1}^N \frac{\lambda_i^x}{\lambda_i} + \frac{1}{2} \ln \left(\prod_{i=1}^N \lambda_i \right)$$

6. $\mathcal{G}$ – dowolne rozkłady normalne.

W tym najbardziej ogólnym przypadku zbiór dzielony będzie na dowolne elipsy [Tabor and Spurek, 2014].



$$\Sigma_{\mathcal{G}}(X) = \Sigma_X$$

$$H^{\times}(X||\mathcal{G}) = \frac{N}{2} \ln(2\pi e) + \frac{1}{2} \ln \det(\Sigma_X)$$

CEC – przykłady

- R script 5

- R script 6

Spis treści

1 Motywacja

2 Entropia

3 Ogólna idea “Cross Entropy Clustering”

4 Zastosowania

- Segmentacja tekstu
- Segmentacja tęczy vs. CEC
 - Cel
 - Algorytm
 - Optymalizacja
 - Rezultat
- Rozpoznawanie “tesktów” z wykorzystaniem CEC-a i jego modyfikacji

Spis treści

1 Motywacja

2 Entropia

3 Ogólna idea “Cross Entropy Clustering”

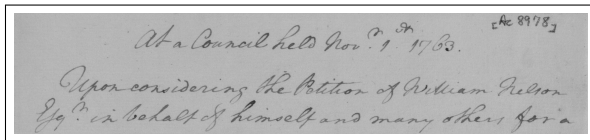
4 Zastosowania

- Segmentacja tekstu

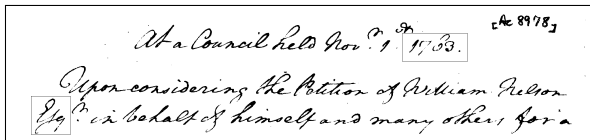
- Segmentacja tęczy vs. CEC

- Rozpoznawanie “tesktów” z wykorzystaniem CEC-a i jego modyfikacji

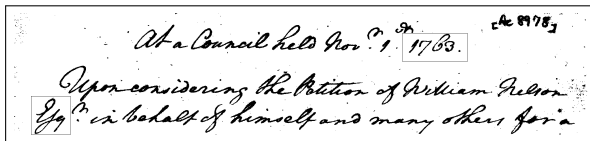
Segmentacja tekstu: Otsu vs. CEC



(a) Zeskanowany tekst.

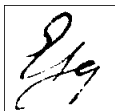
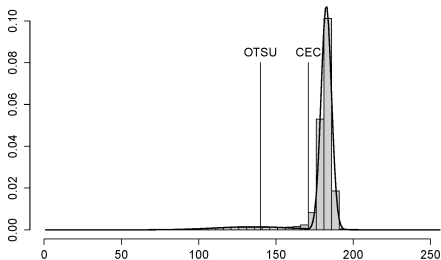


(b) Progowanie Otsu.

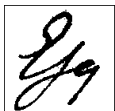


(c) Progowanie CEC [Malik et al., 2016].

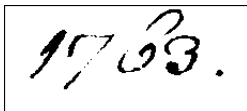
Segmentacja tekstu: Otsu vs. CEC



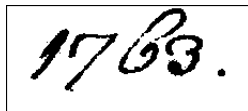
(a) Otsu.



(b) CEC.



(c) Otsu.



(d) CEC.

Spis treści

1 Motywacja

2 Entropia

3 Ogólna idea “Cross Entropy Clustering”

4 Zastosowania

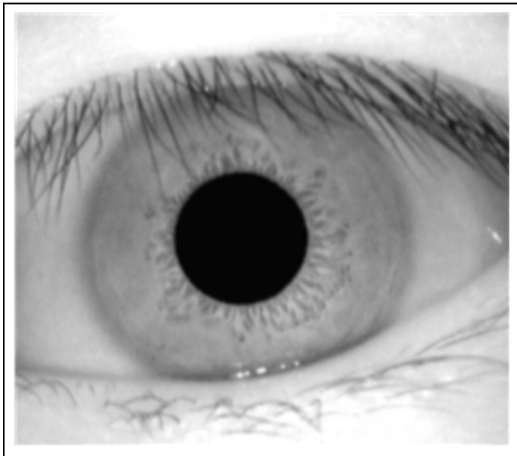
- Segmentacja tekstu

- Segmentacja tęczy vs. CEC

- Rozpoznawanie “tesktów” z wykorzystaniem CEC-a i jego modyfikacji

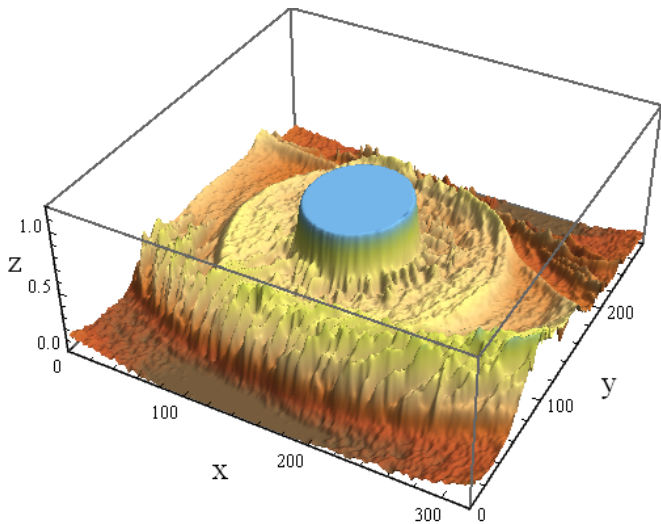
Co chcemy osiągnąć?

x

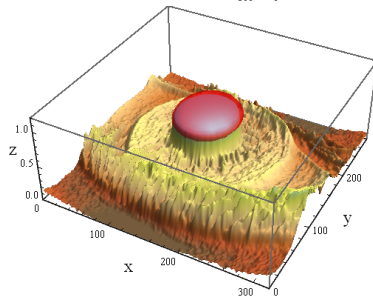
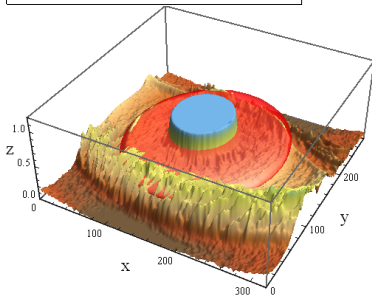
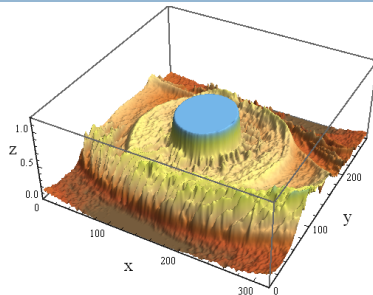
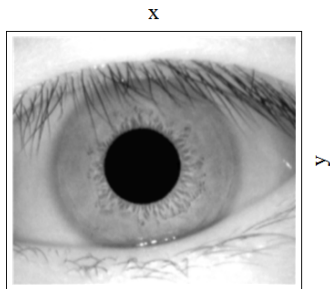


y

Co chcemy osiągnąć?

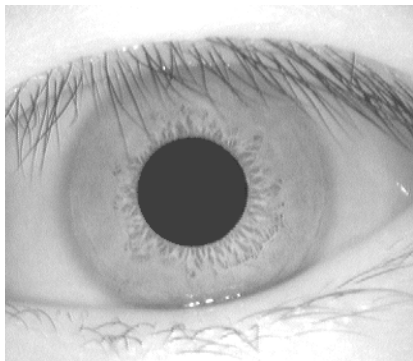


Co chcemy osiągnąć?

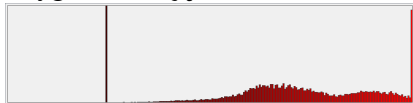


- 1 Maskowanie Gaussem
- 2 Regresja
- 3 Filtrowanie
- 4 Klastrowanie
- 5 Optymalizacja

Maskowanie Gaussem



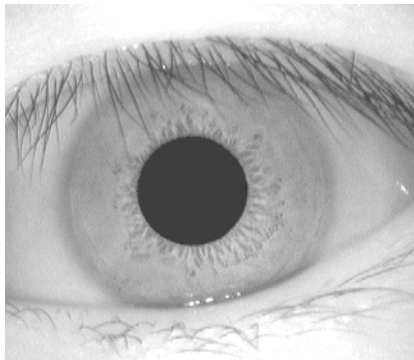
Oryginalne zdjęcie



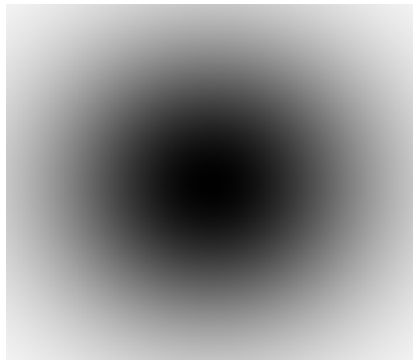
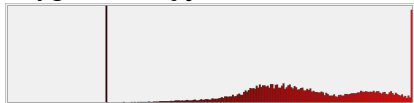
Cel kroku:

Usunięcie zbędnych
elementów/obszarów ze zdjęcia.

Maskowanie Gaussem

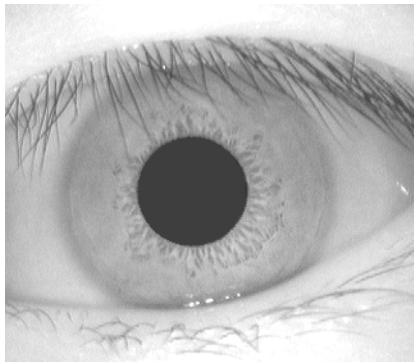


Oryginalne zdjęcie

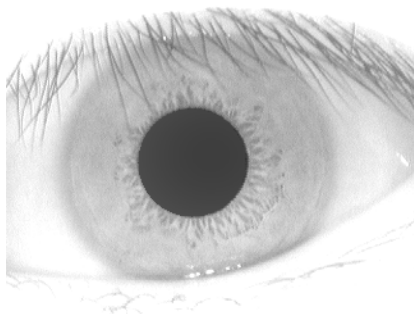
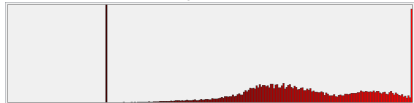


Gauss ze zdjęcia

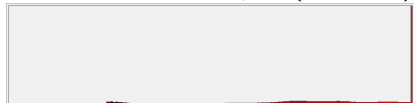
Maskowanie Gaussem



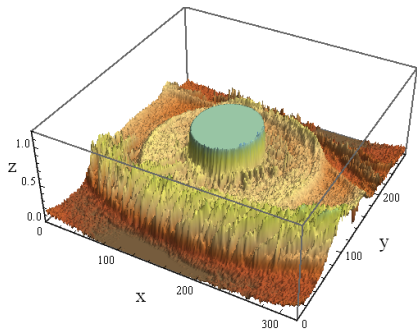
Oryginalne zdjęcie



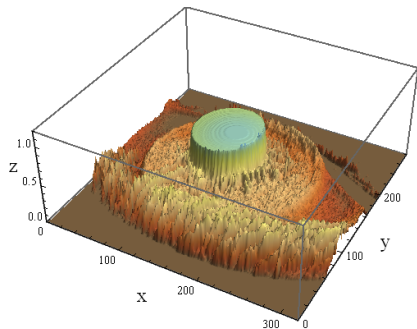
Iris + Gauss ze zdjęcia ($w = 0.3$)



Maskowanie Gaussem

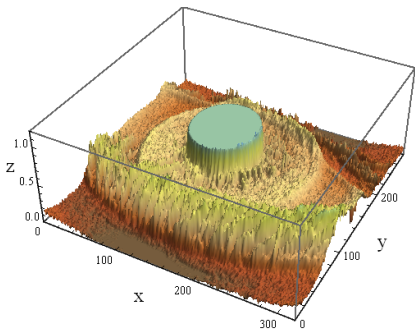


Oryginalne zdjęcie

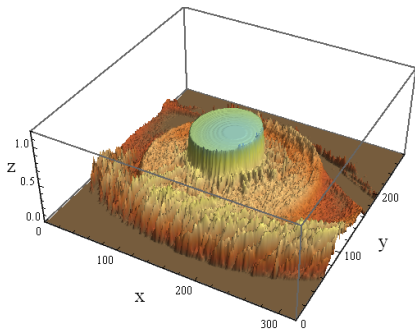


Iris + Gauss ze zdjęcia ($w = 0.3$)

Maskowanie Gaussem



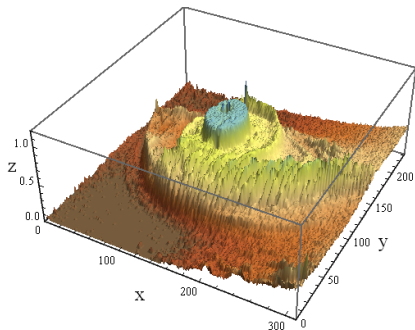
Oryginalne zdjęcie



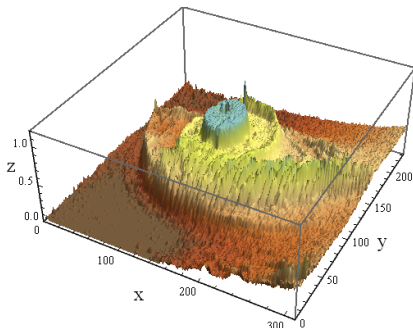
Iris + Gauss ze zdjęcia ($w = 0.3$)

Cel osiągnięty, bo otrzymałem dużo białego koloru na zdjęciu.

Po co tutaj regresja? Co chcę znaleźć?



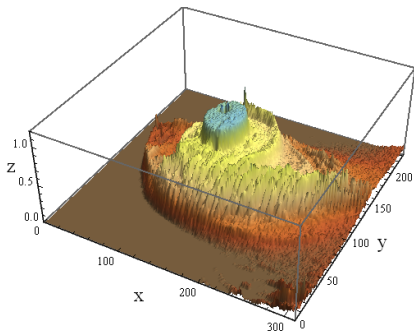
Po co tutaj regresja? Co chcę znaleźć?



Problem

Kolor w źrenicy nie jest równomierny (identyczny).

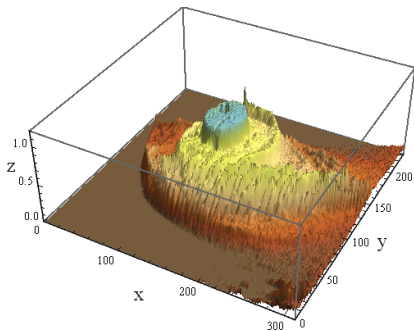
Po co tutaj regresja? Co chcę znaleźć?



Problem

Kolor w źrenicy nie jest równomierny (identyczny). Nawet maskowanie Gaussem nie pomaga, a wręcz psuje jeszcze bardziej.

Po co tutaj regresja? Co chcę znaleźć?



Rozwiązanie

Wyznamy płaszczyznę i względem niej poprawmy zdjęcie.

Jak policzyć regresję?

Problem

Chcemy wyznaczyć płaszczyznę najlepiej dopasowaną do danych.

Jak policzyć regresję?

Problem

Chcemy wyznaczyć płaszczyznę najlepiej dopasowaną do danych.

Rozwiązanie 1

Należy wyliczyć

$$\beta = (X^T X)^{-1} X^T y$$

gdzie

$$X = \begin{pmatrix} 1 & x_1 & y_1 \\ & \vdots & \\ 1 & x_n & y_n \end{pmatrix} \quad y = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix}$$

punkty (x_i, y_i, z_i) to obraz, $\beta = (\beta_0, \beta_1, \beta_2)$ dadzą płaszczyznę.

Jak policzyć regresję?

Problem

Chcemy wyznaczyć płaszczyznę najlepiej dopasowaną do danych.

Rozwiązanie 1

Należy wyliczyć

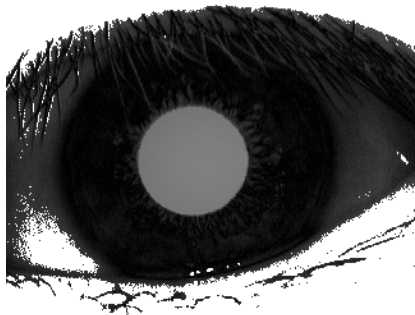
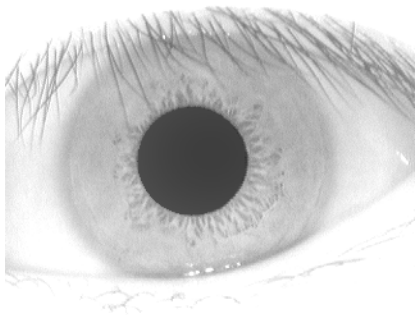
$$\beta = (X^T X)^{-1} X^T y$$

gdzie

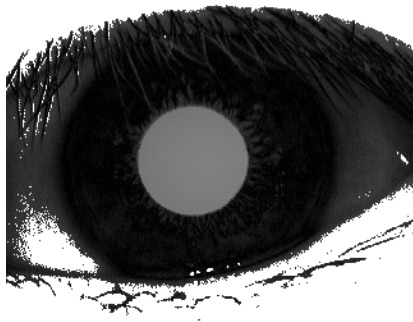
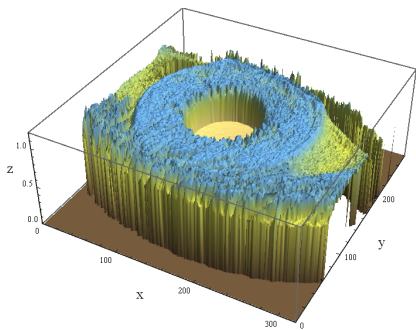
$$X = \begin{pmatrix} 1 & x_1 & y_1 \\ & \vdots & \\ 1 & x_n & y_n \end{pmatrix} \quad y = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix}$$

punkty (x_i, y_i, z_i) to obraz, $\beta = (\beta_0, \beta_1, \beta_2)$ dadzą płaszczyznę.

Regresja



Regresja



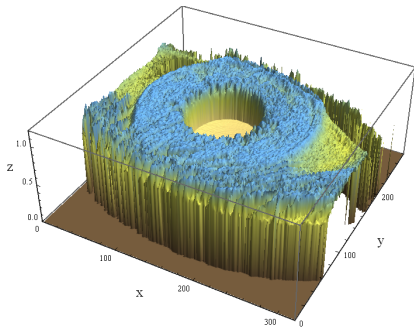
- 1 Wykonujemy po to aby ograniczyć liczbę szczegółów na zdjęciu.

- 1 Wykonujemy po to aby ograniczyć liczbę szczegółów na zdjęciu.
- 2 Wykorzystujemy filtry minimalny (Min) i maksymalny (Max).

Filtrowanie



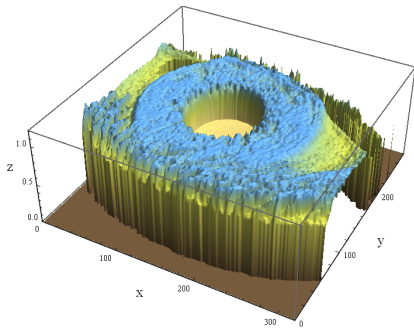
Przed



Filtrowanie



Po



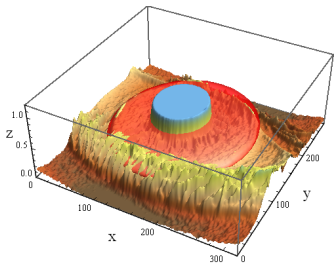
Implementacje CEC-a

- w R: pakiet CEC,
- w Javie: <https://github.com/kmisztal/CEC>.

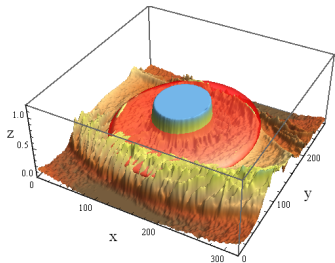


Klastrowanie CEC – dobór modelu

Jaką elipsę chcemy znaleźć?

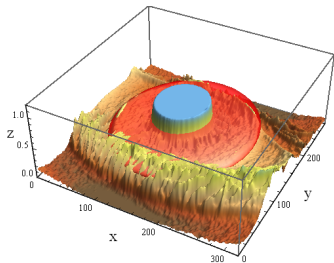


Klastrowanie CEC – dobór modelu



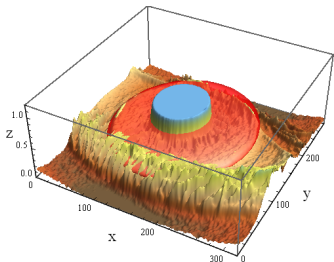
Jaką elipsę chcemy znaleźć?
“spłaszczoną”,
tzn. względem osi OX i OY może mieć
dowolne wymiary, ale chcemy aby w OZ
musi być “chuda”,

Klastrowanie CEC – dobór modelu



Jaką elipsę chcemy znaleźć?
“spłaszczoną”,
tzn. względem osi OX i OY może mieć
dowolne wymiary, ale chcemy aby w OZ
musi być “chuda”,
dodatkowo chcemy aby była ona
równoległą do płaszczyzny OXY .

Klastrowanie CEC – dobór modelu



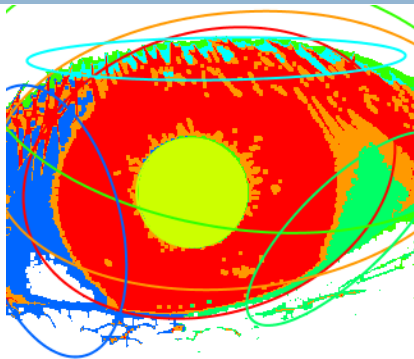
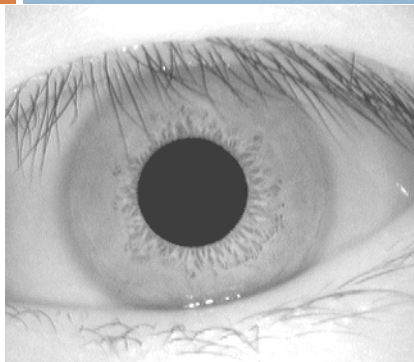
Jaką elipsę chcemy znaleźć?
“spłaszczoną”,
tzn. względem osi OX i OY może mieć
dowolne wymiary, ale chcemy aby w OZ
musi być “chuda”,
dodatkowo chcemy aby była ona
równoległą do płaszczyzny OXY.

Szukana macierz kowariancji:

$$\Sigma_{\varepsilon} = \begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & \varepsilon \end{pmatrix}, \text{ gdzie } \Sigma = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Σ – empiryczna, Σ_{ε} – teoretyczna, ε – małe, ustalone.

Klastrowanie

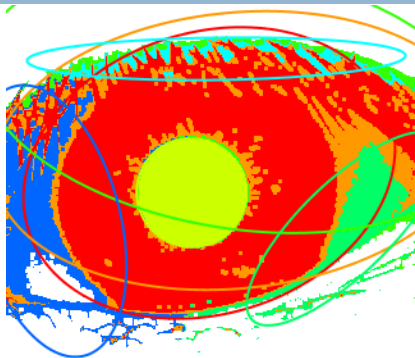


Po CEC-u

Klastrowanie $\Rightarrow$ CEC z 20 klastrami

```
iter.max=50, nstart=24, type=c('special')
```

Klastrowanie



Po CEC-u

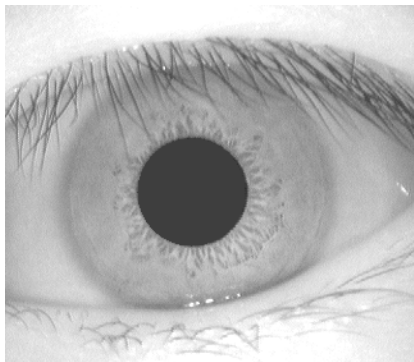
Klastrowanie $\Rightarrow$ CEC z 20 klastrami

```
iter.max=50, nstart=24, type=c('special')
```

W tym kroku:

- wybierzemy klastry tęczy i żrenicy,
- usuniemy “zbędne” punkty z klastrów.

Optymalizacja – dwa klastry



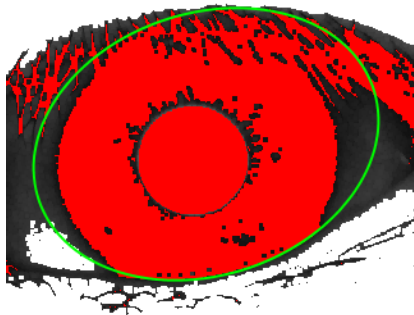
Dwa klastry o najmniejszej wariancji w OZ.

Twierdzenie ([Misztal and Tabor, 2013])

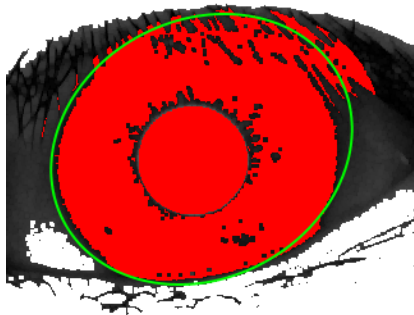
Dla rozkładu jednostajnego na elipsie $E \subset \mathbb{R}^N$ o średniej m_E i macierzy kowariancji Σ_E zachodzi

$$E = \mathcal{B}_{\Sigma_E} \left(m_E, \sqrt{N+2} \right). \quad (8)$$

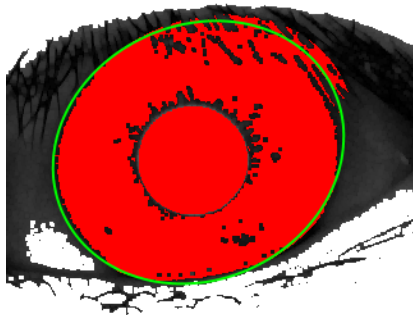
$$\mathcal{B}_{\Sigma_E}(\mu_E, r) = \{x \in \mathbb{R}^N : (x - \mu_E)^T \Sigma_E^{-1} (x - \mu_E) \leq r^2\}.$$



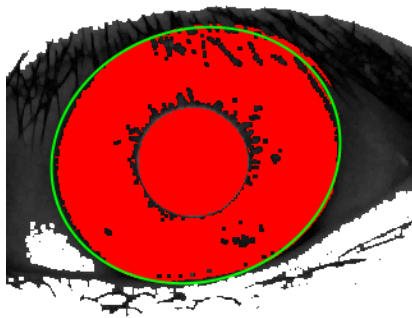
Do klastra tęczówki dodano punkty z klastra źrenicy – cały zbiór będziemy optymalizować.



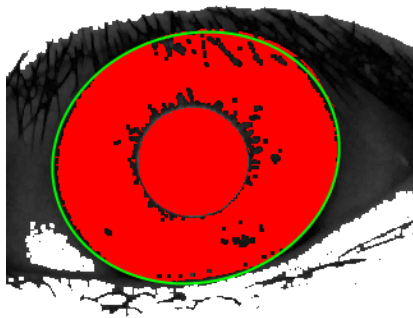
Do klastra tęczówki dodano punkty z klastra źrenicy – cały zbiór będziemy optymalizować.



Do klastra tęczówki dodano punkty z klastra źrenicy – cały zbiór będziemy optymalizować.

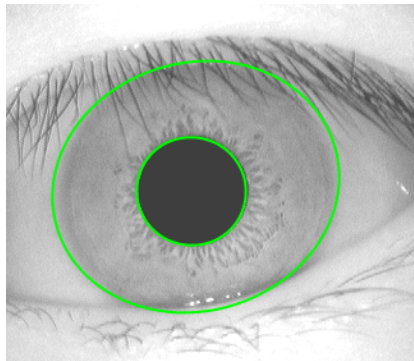
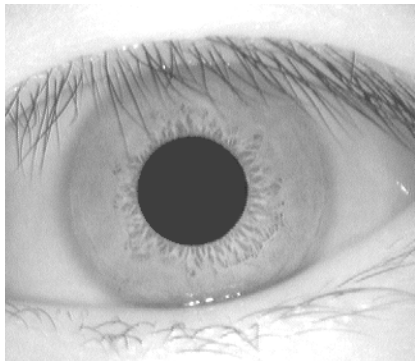


Do klastra tęczówki dodano punkty z klastra źrenicy – cały zbiór będziemy optymalizować.



Do klastra tęczówki dodano punkty z klastra źrenicy – cały zbiór będziemy optymalizować.

Wynik końcowy



- 1 Maskowanie Gaussem $\Rightarrow$ CEC z jednym klastrem
`iter.max=1, nstart=1, type=c('all')`
- 2 Regresja
- 3 Filtrowanie $\Rightarrow$ Min, Max x n
- 4 Klastrowanie $\Rightarrow$ CEC z 20 klastrami
`iter.max=50, nstart=24, type=c('special')`
- 5 Optymalizacja $\Rightarrow$ usuwanie wystających części

Spis treści

1 Motywacja

2 Entropia

3 Ogólna idea “Cross Entropy Clustering”

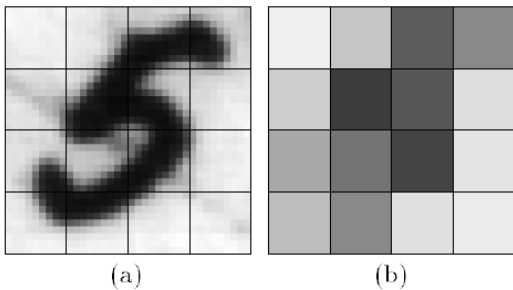
4 Zastosowania

- Segmentacja tekstu

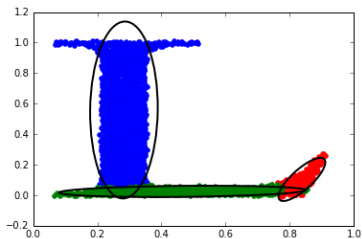
- Segmentacja tęczy vs. CEC

- Rozpoznawanie “tesktów” z wykorzystaniem CEC-a i jego modyfikacji

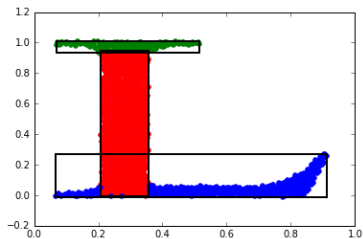
Zonning – podstawowy algorytm



“Prostokątny” CEC

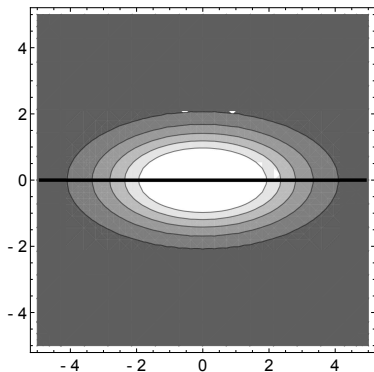


(a) CEC

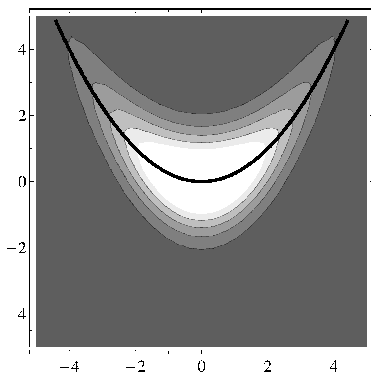


(b) Uniform-CEC

“Wygięty” CEC

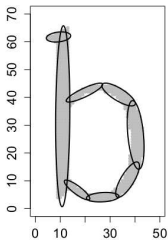


(a) Poziomice rozkładu normalnego

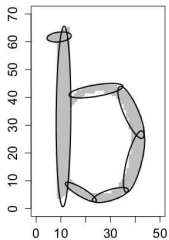


(b) Poziomice wygitego rozkładu normalnego

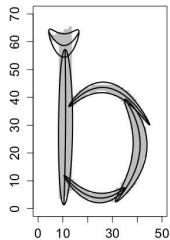
“Wygięty” CEC



(a) GMM, $k = 8$



(b) CEC, $k = 7$



(c) afCEC, $k = 5$
[Spurek et al., 2017]

Dziękuję za uwagę.

Bibliography I

- [0] Jain, A. K., Ross, A., and Prabhakar, S. (2004).
An introduction to biometric recognition.
Circuits and Systems for Video Technology, IEEE Transactions on, 14(1):4–20.
- [0] Malik, M., Spurek, P., and Tabor, J. (2016).
Cross-entropy based image thresholding.
Schedae Informaticae, 24:21–29.
- [0] Misztal, K., Saeed, E., Saeed, K., Spurek, P., and Tabor, J. (2016).
Cross entropy clustering approach to iris segmentation for biometrics purpose.
Schedae Informaticae, 24:31–40.
- [0] Misztal, K. and Tabor, J. (2013).
Mahalanobis distance-based algorithm for ellipse growing in iris preprocessing.
In *Computer Information Systems and Industrial Management*, volume 8104,
pages 158–167. Lecture Notes in Computer Science, Springer Berlin Heidelberg.
- [0] Spurek, P. (2017).
General split gaussian cross-entropy clustering.
Expert Systems with Applications, 68:58–68.
- [0] Spurek, P., Kamieniecki, K., Tabor, J., Misztal, K., and Śmieja, M. (2016).
R package cec.
Neurocomputing.

Bibliography II

- [0] Spurek, P., Tabor, J., and Byrski, K. (2017).
Active function cross-entropy clustering.
Expert Systems with Applications, 72:49–66.
- [0] Spurek, P., Tabor, J., and Zając, E. (2013).
Detection of disk-like particles in electron microscopy images.
In *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, pages 411–417. Springer International Publishing.
- [0] Tabor, J. and Misztal, K. (2013).
Detection of elliptical shapes via cross-entropy clustering.
In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 656–663. Springer Berlin Heidelberg.
- [0] Tabor, J. and Spurek, P. (2014).
Cross-entropy clustering.
Pattern Recognition, 47(9):3046–3059.
- [0] Tomczyk, A., Spurek, P., Podgórski, M., Misztal, K., and Tabor, J. (2016).
Detection of elongated structures with hierarchical active partitions and cec-based image representation.
In *Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015*, pages 159–168. Springer International Publishing.