Rozpoznávání tváří I

Vojtěch Franc
Centrum strojového vnímání, ČVUT FEL Praha

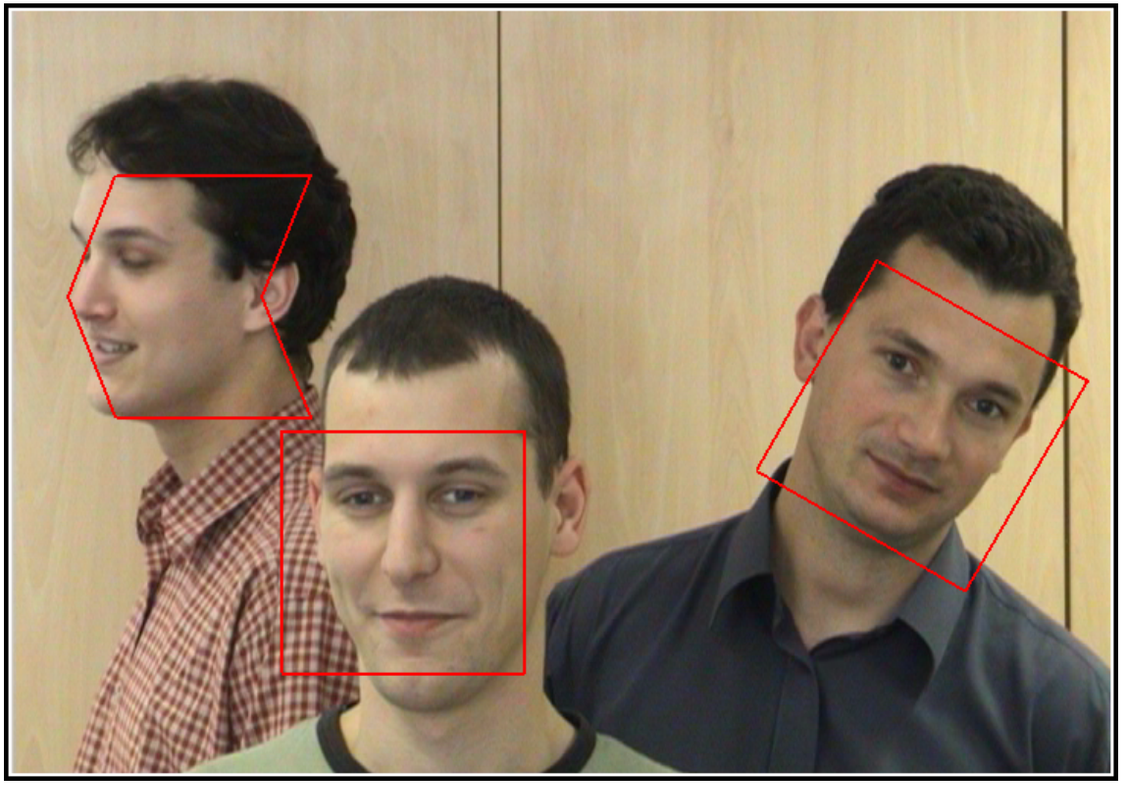
Biometrie ZS 2016

Osnova:

- Příklady úloh v rozpoznávání tváří: detekce, verifikace, vyhledávání, odhad věku, ...
- Metriky pro měření přesnosti
- Detekce tváří

Úlohy rozpoznávání tváří: Detekce

- Úloha: lokalizovat tváře ve vstupním obrázku.
- Výstupem detektoru je pozice, velikost a natočení nalezených tváří.
- Aplikace: nezbytný první krok při rozpoznávání tváří, automatické zaostřování v digitálních kamerách, ...



Metriky pro měření přesnosti detektorů tváří

- Testovací sada: databáze obrázků s manuálně anotovanými pozicemi tváří
- Správná/falešná detekce definovaná pomocí "Intersection Over Union ratio"

$$\text{IOU}(A, B) = \frac{A \cap B}{A \cup B} \in [0, 1]$$

kde A a B jsou oblasti (množiny pixelů), kde se nalézá nalezený/anotovaný objekt.

- $\text{IOU}(A, B) \geq \theta$ (např. $\theta = 0.7$) implikuje správnou detekci a $\text{IOU}(A, B) < \theta$ falešnou detekci.
- True Positive Rate

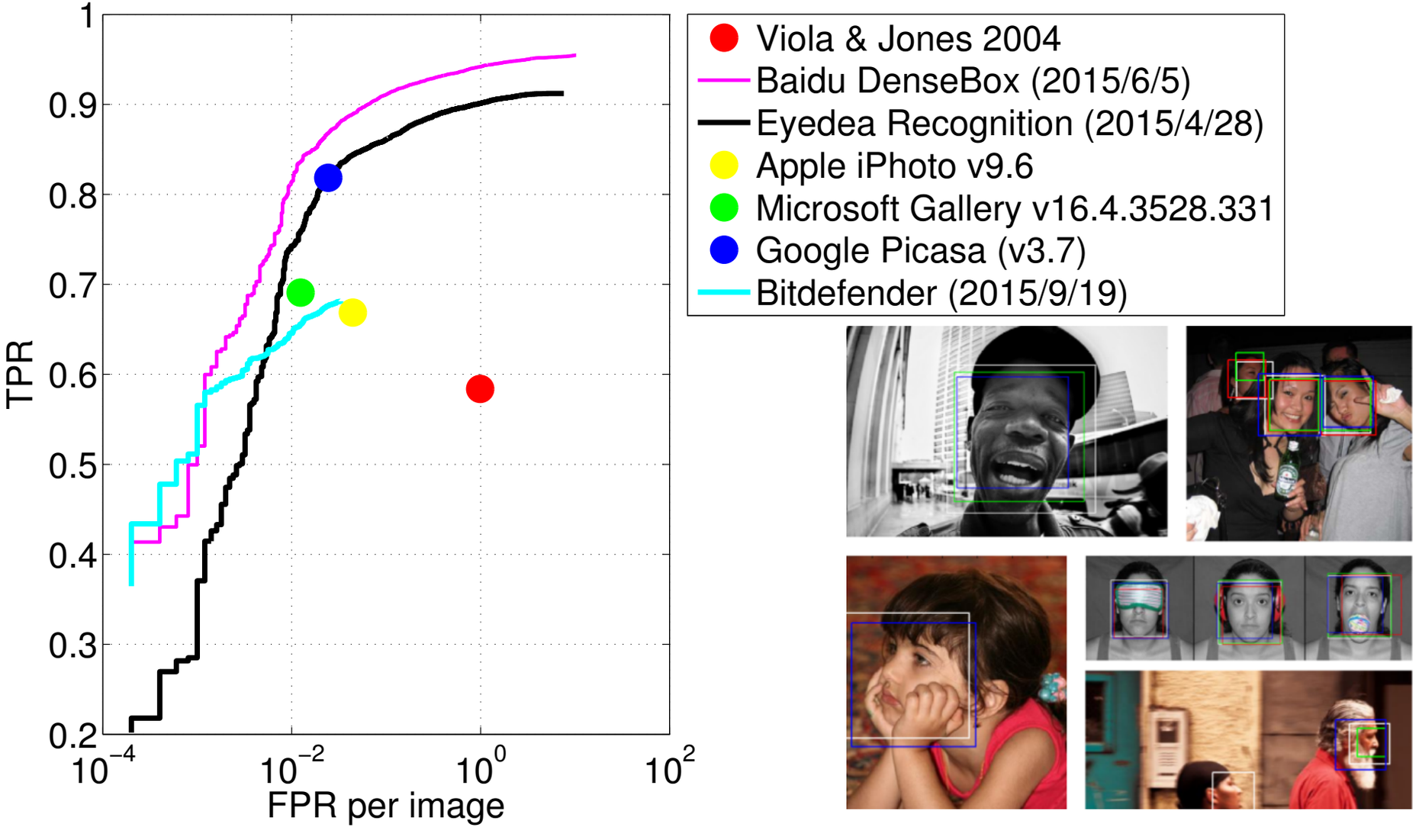
$$\text{TPR} = \frac{\text{počet správných detekcí}}{\text{počet všech anotovaných tváří}} \in [0, 1]$$

- False Positive Rate per image

$$\text{FPR per image} = \frac{\text{počet falešných detekcí}}{\text{počet všech obrázků}} \in [0, \infty)$$

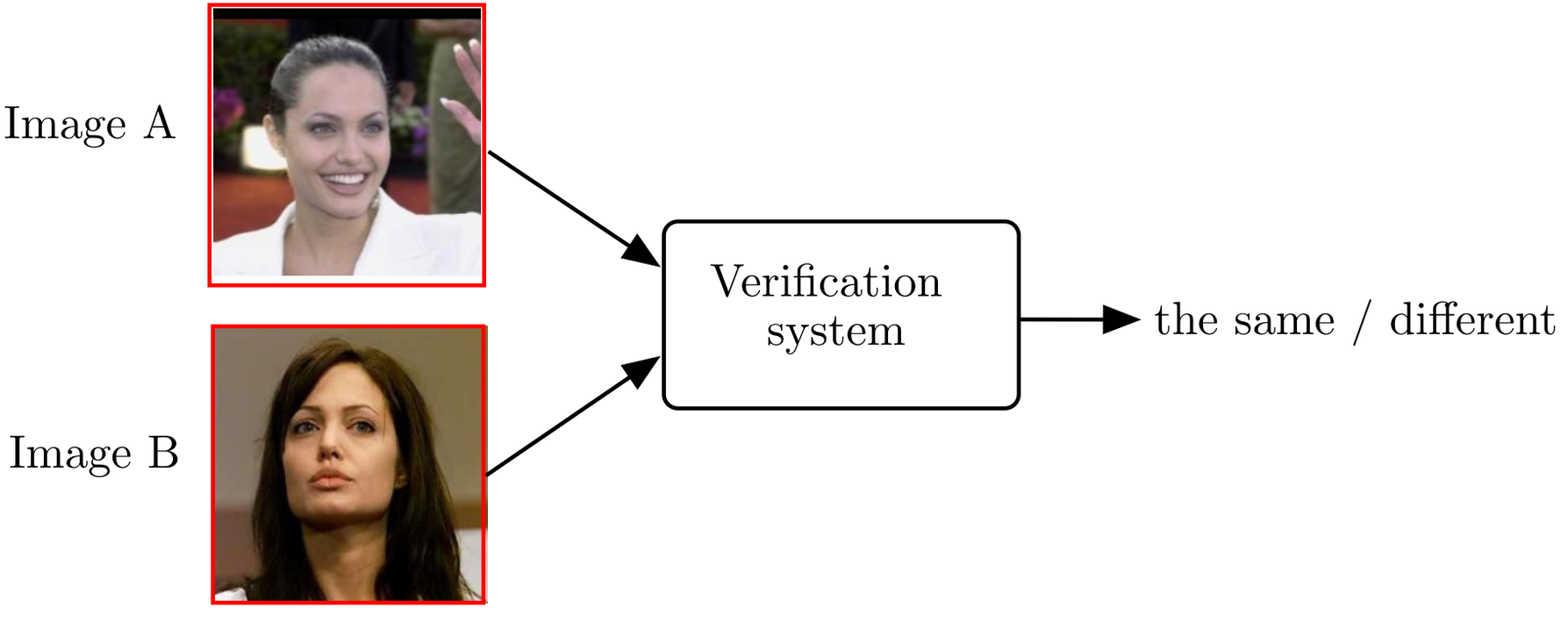
Přesnost současných detektorů na reálných obrázcích

- Fine-grained Evaluation on Face Detection in The Wild [Yang et al 2015] je benchmark pro měření přesnosti detektorů tváří.
- 5,250 obrázků s 11,931 tvářemi (velký rozsah úhlu pohledu) stáženými z Internetu.



Úlohy rozpoznávání tváří: Verifikace identity

- Úloha: pro zadanou dvojici obrázků tváří ověřit zdali se jedná o stejnou identitu.
- Výstup: binární rozhodnutí "stejná osoba/dvě různé osoby" popřípadě doplněné konfidencí odhadu.
- Aplikace: přístup do kontrolovaných prostor (např. průchod letištěm), odemykaní mobilních telefonů,...



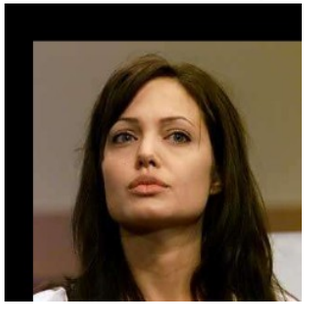
LFW databáze pro měření přesnost verifikačních metod

- Labeled Faces in the Wild [Huang et al 2008] je sada obrázků a definice testovacích protokolů pro měření přesnosti verifikačních metod.
- 13,233 fotografií známých osobností (herci, politici, sportovci, ...) v rozlišení 250×250 pixelů
- Testovací protokol definuje testovací a trénovací dvojice obrázků:

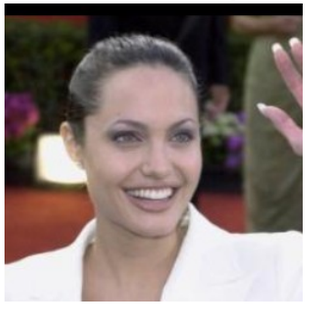
$$\{(A^1, B^1, y^1), \ldots, (A^m, B^m, y^m)\}$$

kde A^i a B^i jsou obrázky a $y^i \in \{+1/-1\}$ je label značící pozitivní/negativní příklad.

$y = +1$ (stejná identita)

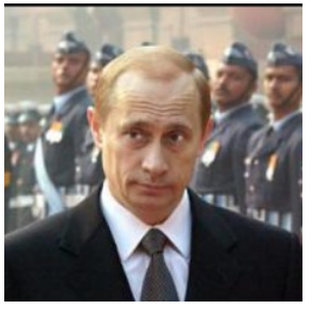


A

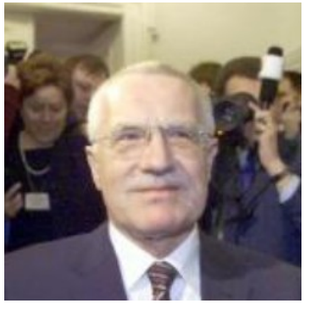


B

$y = -1$ (různé identity)



A



B

- Verifikační algoritmus je funkce $f(\text{obrázek A, obrázek B}) \rightarrow \{+1, -1\}$.

Metriky pro měření verifikačních algoritmů

- False Positive Rate

$$\text{FPR} = \frac{\text{počet negativních příkladů klasifikovaných jako pozitivní}}{\text{počet všech negativních příkladů}}$$

- Flase Negative Rate

$$\text{FNR} = \frac{\text{počet pozitivních příkladů klasifikovaných jako negativní}}{\text{počet všech pozitivních příkladů}}$$

- Equal Error Rate je pracovní bod kdy platí $\text{FPR} = \text{FNR}$

- True Positive Rate

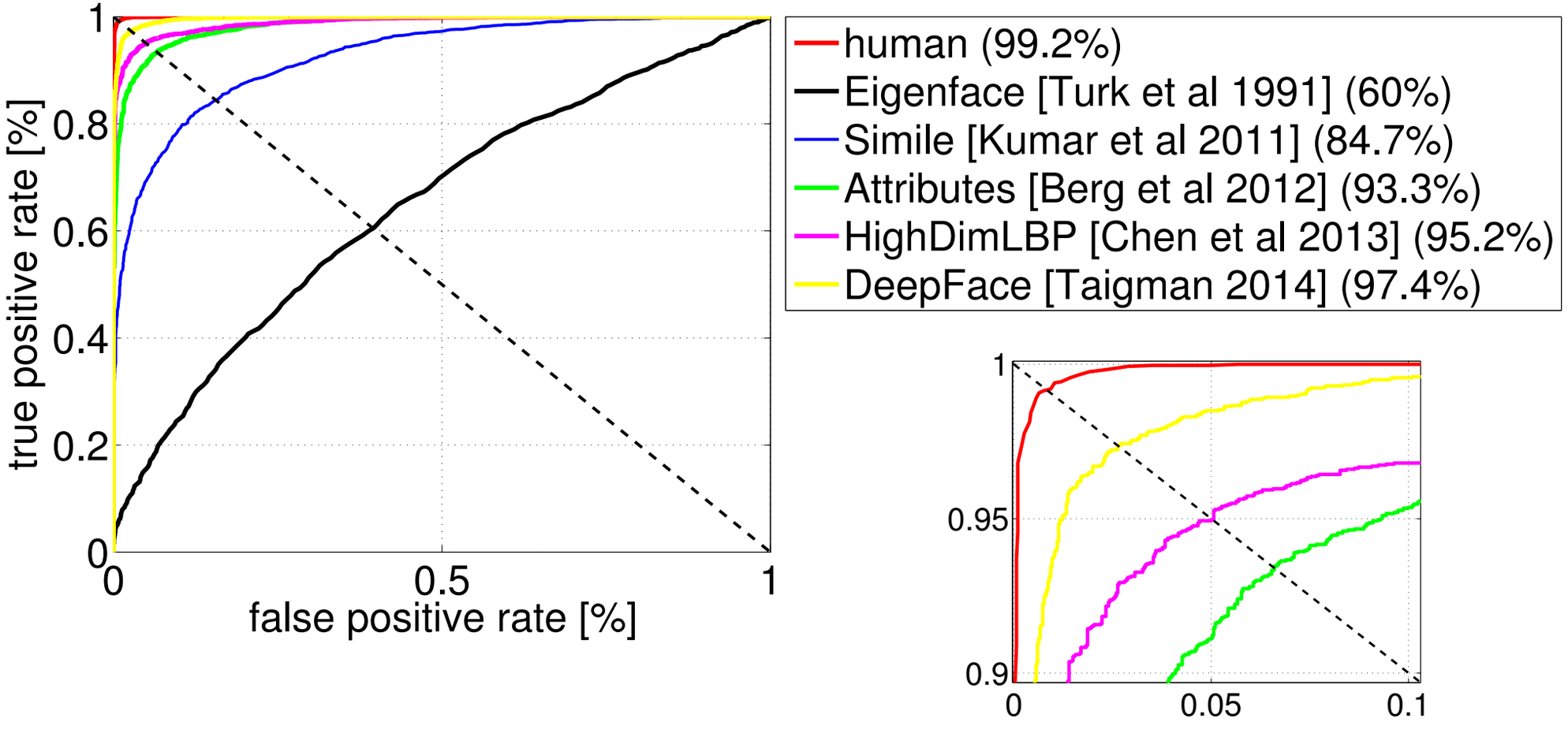
$$\text{TPR} = 1 - \text{FNR} = \frac{\text{počet pozitivních příkladů klasifikovaných jako pozitivní}}{\text{počet všech pozitivních příkladů}}$$

Přesnost verifikačních metod na LFW databázi

- Verifikační stroje často rozhodují na základě podobnostní funkce $s\colon I \times I \to \Re$, pomocí které se definuje rozhodovací funkce pro verifikaci

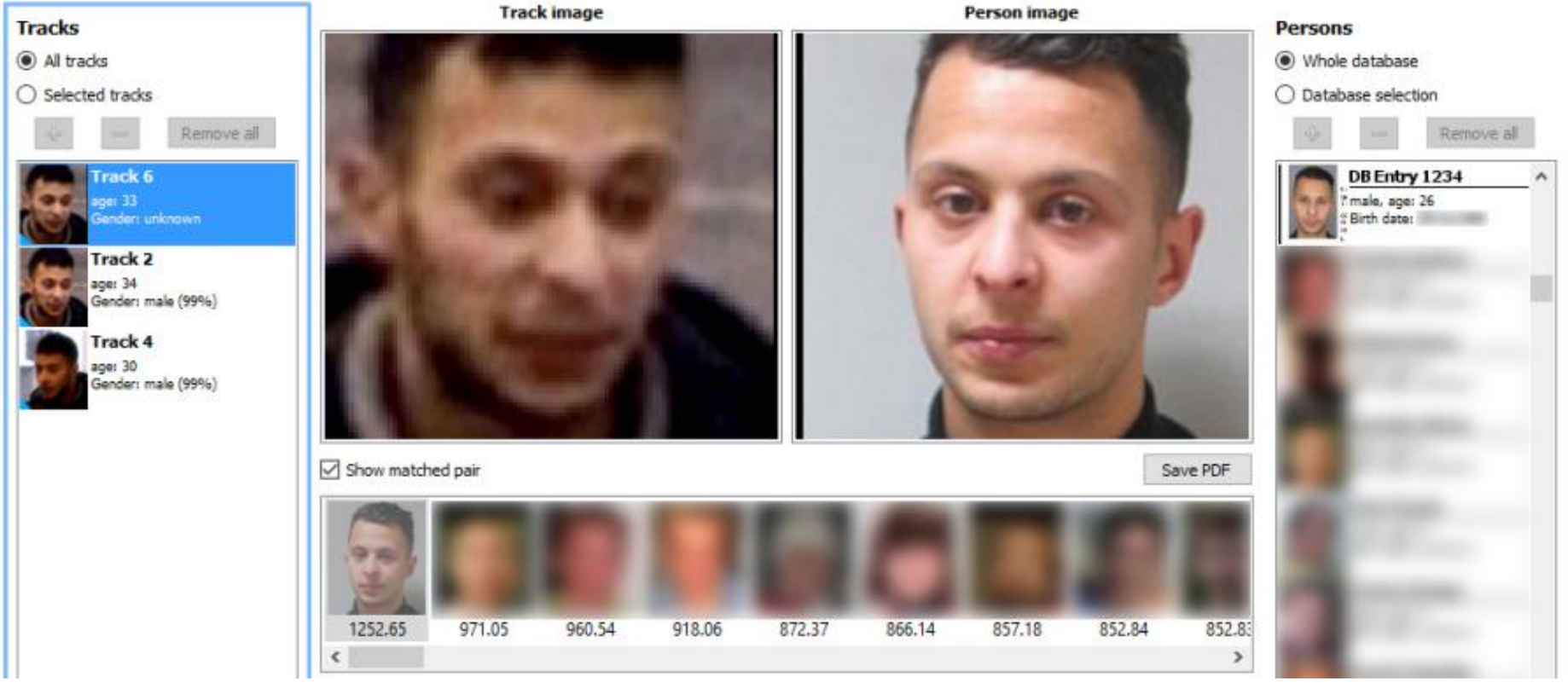
$$f_\theta(\text{obrázek A, obrázek B}) = \begin{cases} +1 & \text{pokud} \quad s(\text{obrázek A, obrázek B}) \geq \theta \\ -1 & \text{pokud} \quad s(\text{obrázek A, obrázek B}) < \theta \end{cases}$$

- FPR(θ) a TPR(θ) jako funkce θ pro vybrané verifikační metody ohodnocené na LFW:



Úlohy rozpoznávání tváří: Vyhledávání (face retieval)

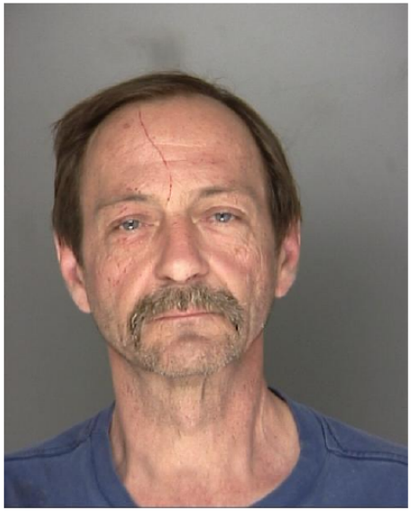
- Úloha: pro zadanou tvář nalézt v databázi s M tvářemi N nejpodobnějších.
- Výstupem je seřazený seznam N nejpodobnějších tváří.
- Aplikace: zpracování záznamů z dohledových kamer, vyhledávání v databázi zadržovaných osob, detekce duplikátů v databázi, "tagování" filmů, ...



Eyedea Recognition www.eyedea.cz

Přesnost vyhledávání tváří podle FRVT2013

- Face Recognition Vendor Test 2013: soutěž organizovaná National Institue of Standards and Technology (NIST) s cílem vyhodnotit existující technologie rozpoznávání tváří.
- Testováno na velké databázi (1.6 miliónů) policejních fotografii zadržených osob (magshot) a obrázků z běžné web-camery.



(a) Good quality mugshot

(b) Poor quality webcam

	Mugshots		WebCam
	R1	R50	R1
NEC	4.1 %	2.6%	11.3 %
Morpho	9.1 %	7.1%	29.8 %
Toshiba	10.7 %	5.7%	23.7 %
Cognitec	13.6 %	8.4%	57.6 %

- R1 ... pravděpodobnost, že hledaná identita (obsažená v databázi) nebude na prvním místě vráceného seznamu.
- R50 ... pravděpodobnost, že hledaná indentida nebude v seznamu délky 50.

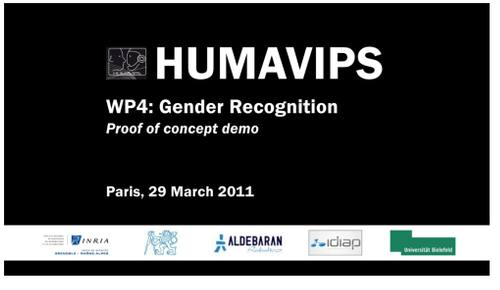
Obrázek a výsledky převzaty z *Grother et al: Face Recognition Vendor Test, NIST Interagency report 8009. May 26, 2014.*

Úlohy rozpoznávání tváří: Odhadování věku a pohlaví

- Úloha: na základě obrázku tváře odhadnout věk či pohlaví zobrazené osoby
- Aplikace: povolení přístupu na základě věku (např. automaty na cigarety v Japonsku), demografické průzkumy (audience measurement systems), cílená reklama (např. na benzinových pumpách Tesco v Anglii), ...

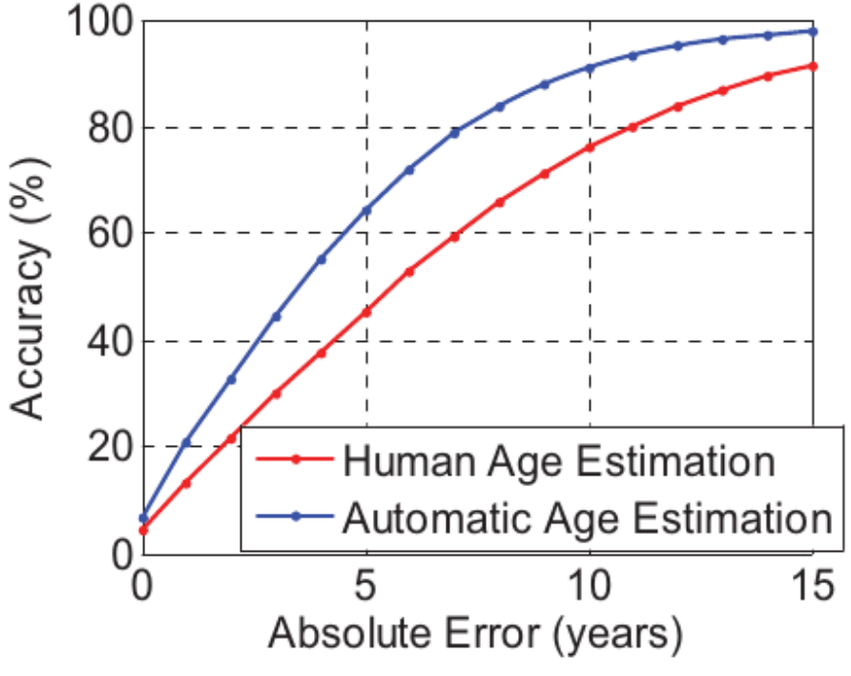
Automat na cigarety ověřující věk pomocí zabudované kamery

"Aplikace" v robotice



Odhadování věku: stroj versus člověk

- Test na PSCO databázi 10,036 policejních fotografii s rovnoměrným zastoupením věku v rozmezí 17 až 68 let.
- Věk 2,200 náhodně vybraných tváří odhadnut 10. lidmi (Amazon Mechanical Turk service); maximální a minimální odhad se nepoužije a bere se aritmetický průměr zbývajících 8. odhadů.
- Přesnost odhadu měřena pomocí absolutní odchylky odhadnutého a skutečného věku.

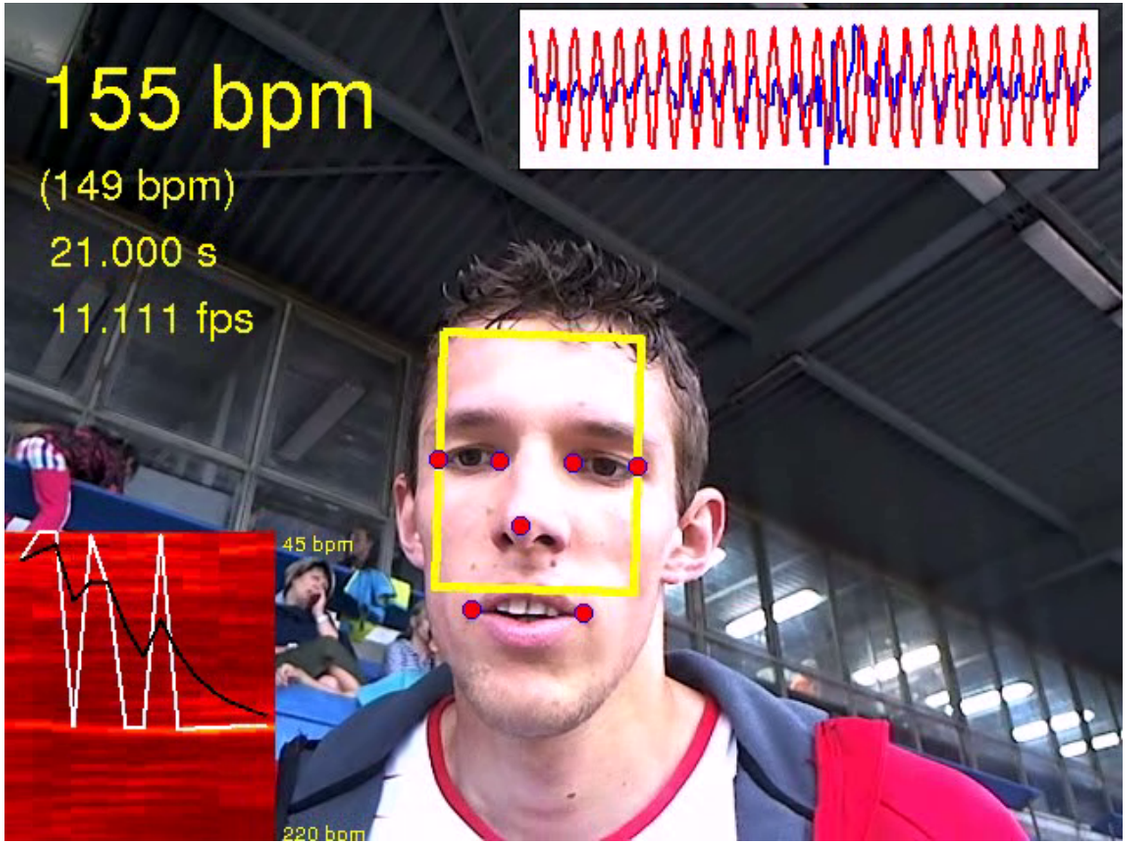


	MAE
Human	7.2
Machine	5.1

Výsledky převzaty z článku *Han et al: Age estimation from Face Images: Human vs. Machine Performance. Inter. Conf. on Biometrics. 2013.*

Úlohy rozpoznávání tváří: Odhad tepové frekvence

- Úloha: odhadnout tepovou frekvenci na základě videa sledovaného subjektu.
- Výstupem systému je tepová frekvence.
- Příklad úlohy, kterou člověk není schopný řešit bez pomoci stroje.



Rozpoznávání tváří: porovnání s ostatními biometrickými metodami

Silné stránky:

- Nevyžaduje kooperující subjekt narozdíl od rozpoznávání otisků prstů, duhovky atd.
- Bezkontaktní snímač.
- Sensory pro měření (např. standardní kamery) jsou levné a masově rozšířené.
- Lze využít pro širokou třídu úloh (rozpoznávání identity, věku, pohlaví, emocí, rasy, tepu...).

Rozpoznávání tváří: porovnání s ostatními biometrickými metodami

Silné stránky:

- Nevyžaduje kooperující subjekt narozdíl od rozpoznávání otisků prstů, duhovky atd.
- Bezkontaktní snímač.
- Sensory pro měření (např. standardní kamery) jsou levné a masově rozšířené.
- Lze využít pro širokou třídu úloh (rozpoznávání identity, věku, pohlaví, emocí, rasy, tepu...).

Slabiny:

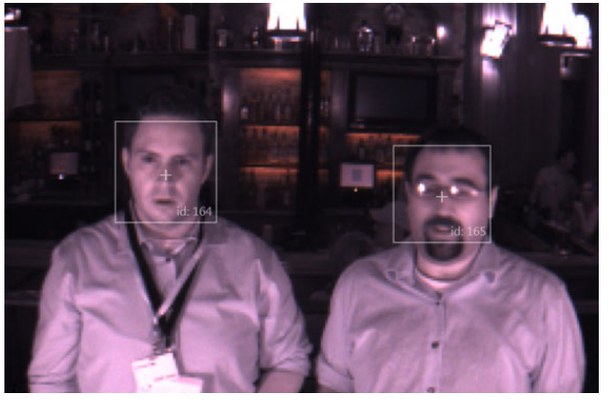
- Pro odhad identity méně přesné (přestože se již blížící přesnosti člověka) v porovnání s technologiemi jako rozpoznávání otisků nebo duhovky.
- Pro dosažení vysoké efektivity je třeba zajistit kontrolované podmínek či specializovaný hardware.

Proč je rozpoznávání tváří těžké?

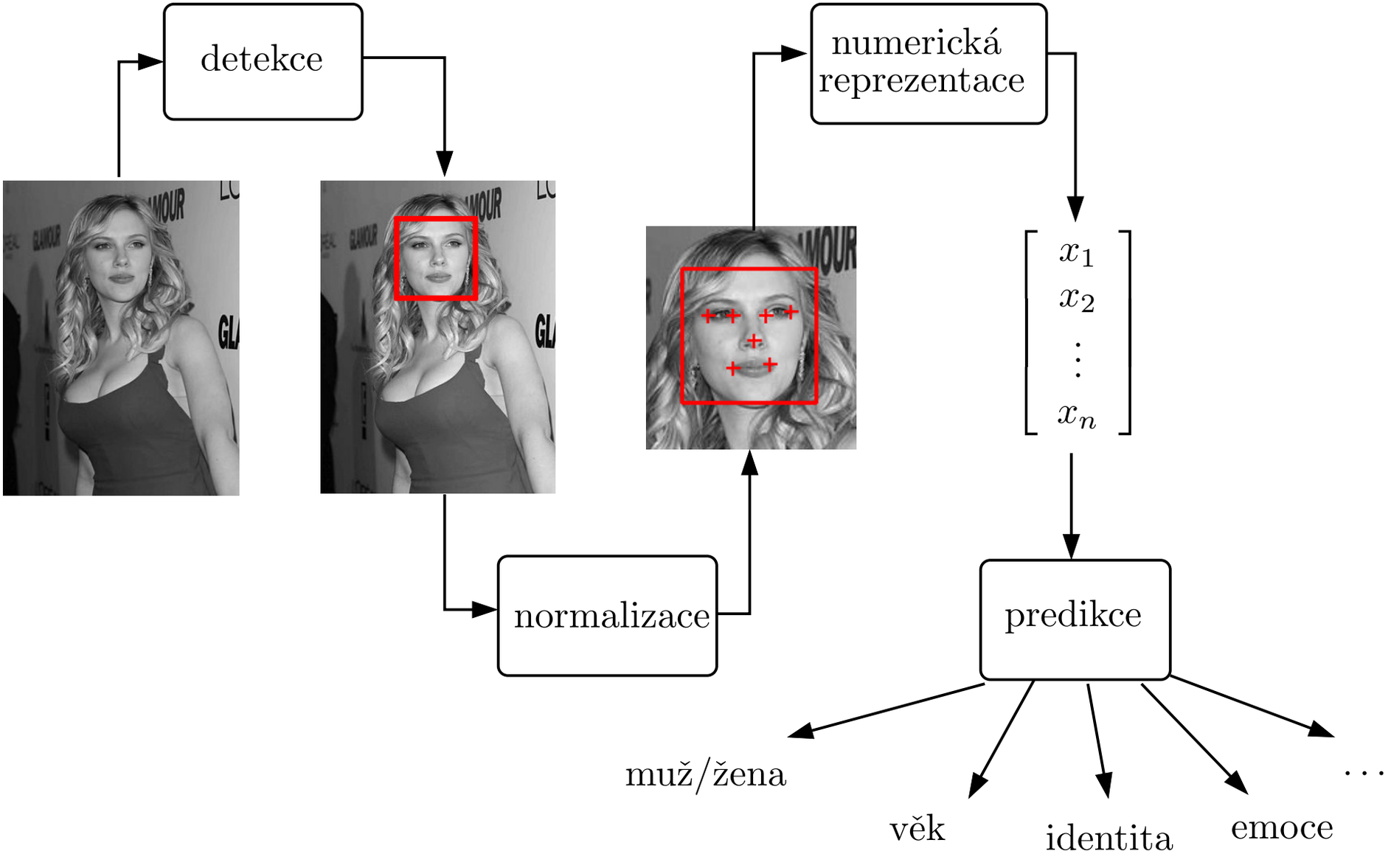
- Signál ze senzoru je vysokodimenziální. Např. 10,000 dimenzionální signál z kamery s rozlišením 100×100 pixelů.
- V signálu pocházejícího z měření patřících do jedné třídy je velká variabilita, způsobená:
  - změnou pozice, meřítka, rotace (roll, pan, tilt)
  - změnou osvětlení
  - změnou výrazu tváře (úsměv, smutek, neutrální, ...)
  - zákryty (brýle, pokrývka hlavy)
  - změna účesu, make up, stárnutím ...

Funkčnost systému rozpoznávání tváří je ovlivňena mnoha faktory

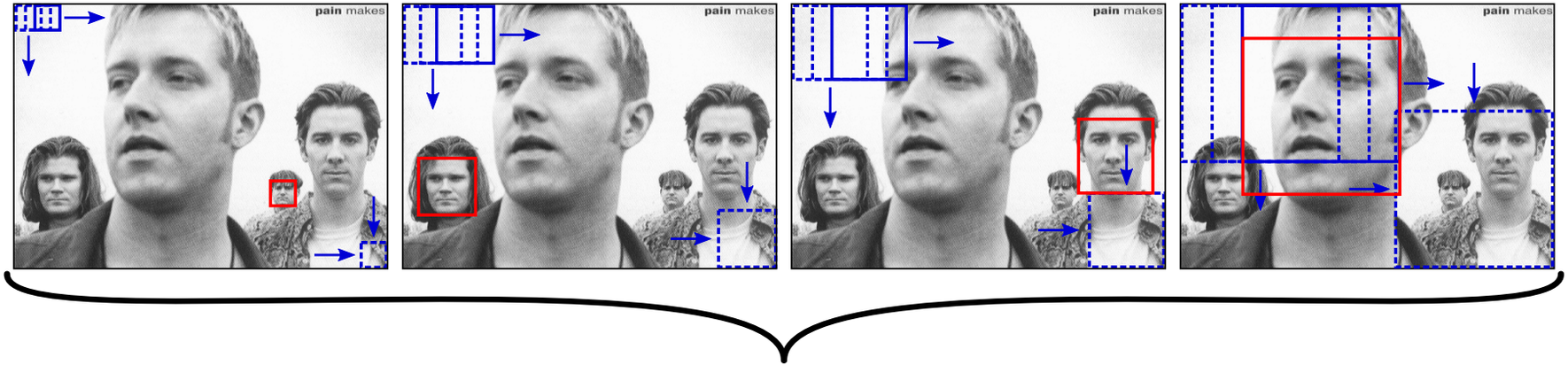
- Použitý snímací senzor:
  - kamera pracující ve viditelném spektru
  - infračervená kamera (s IR přísvícením)
  - stereo kamera
  - 3D skener
- Světelné podmínky (venkovní/vnitřní prostor, stíny, ...)
- Rozlišení obrazu a použitá komprese
- Statický obrázek / video sekvence
- Spolupracující / nespolupracující subjekt
- . . .

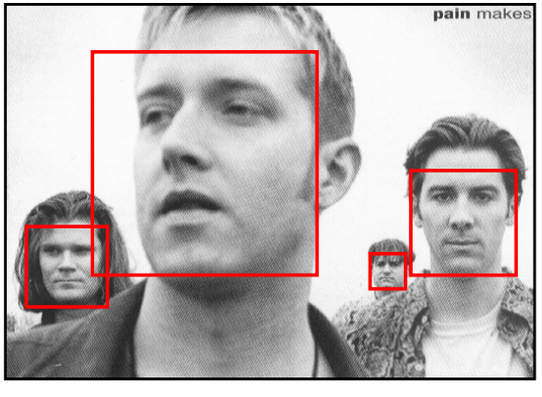


Stavební bloky typického systému pro rozpoznávání tváří



Detektor tváří





Těžký predikční problém se převede na mnoho jednodušších H : Obrázek $\rightarrow \{\text{tvář}, \text{netvář}\}$

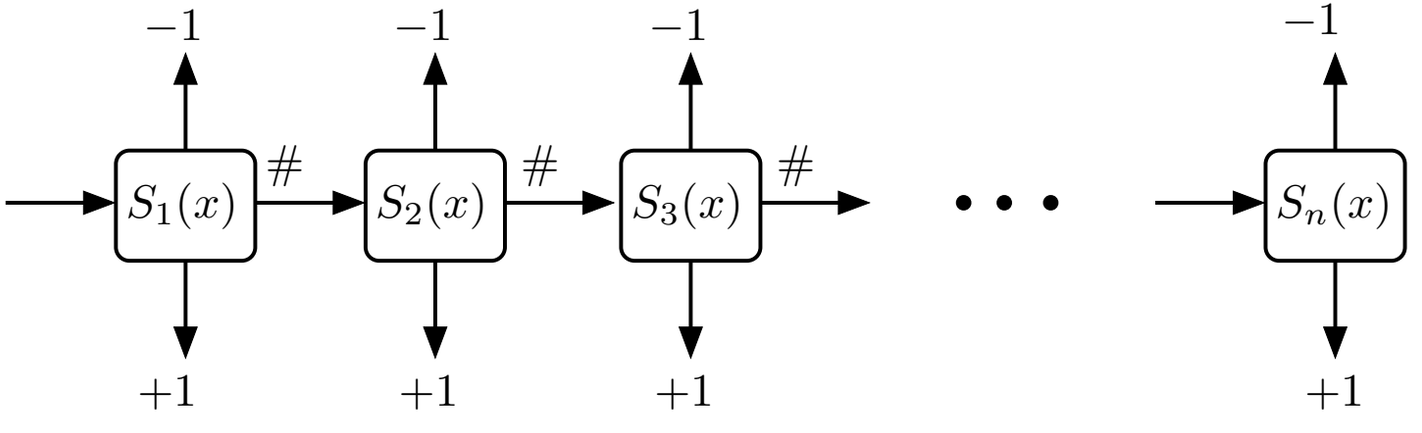
Klasifkátor tvář/netvář pro detekci tváří

- **Požadavky na klasifikátor:** extrémně přesný a extrémně rychlý.
  - Např. nalezení tváře velikosti 24×24 pixelů v obrázeku 640x480 vyžaduje provést $(640 - 24 + 1) \cdot (480 - 24 + 1) = 281,969$ binárních klasifikací.
  - Rychlost: Při 10 FPS musí jedna klasifikace trvat nanejvýš $0.34\mu s$
  - Přesnost: $\mathrm{FP} \to 0\%$ (chybné detekce), $\mathrm{TP} \to 100\%$ (detekované tváře)

Klasifkátor tvář/netvář pro detekci tváří

- **Požadavky na klasifikátor:** extrémně přesný a extrémně rychlý.
  - Např. nalezení tváře velikosti 24×24 pixelů v obrázeku 640x480 vyžaduje provést $(640 - 24 + 1) \cdot (480 - 24 + 1) = 281,969$ binárních klasifikací.
  - Rychlost: Při 10 FPS musí jedna klasifikace trvat nanejvýš $0.34\mu s$
  - Přesnost: $\mathrm{FP} \to 0\%$ (chybné detekce), $\mathrm{TP} \to 100\%$ (detekované tváře)
- **Řešení:** sekvenční rozhodovací pravidlo složené z jednoduchých rychlých klasifikátorů

$$S_t(x) = \begin{cases} +1 & \text{(tvář)} & f_t(x) \geq \theta_t^A \\ -1 & \text{(netvář)} & f_t(x) \leq \theta_t^B \\ \# & \text{(nevím)} & \theta_t^B < f_t(x) < \theta_t^A \end{cases} \qquad f_t(x) = \sum_{i=1}^{t} \alpha_i h_i(x)$$



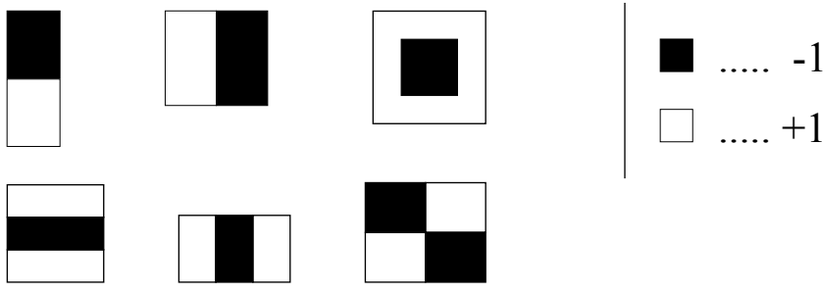
AdaBoost: sestroj silný klasifikátor ze slabých

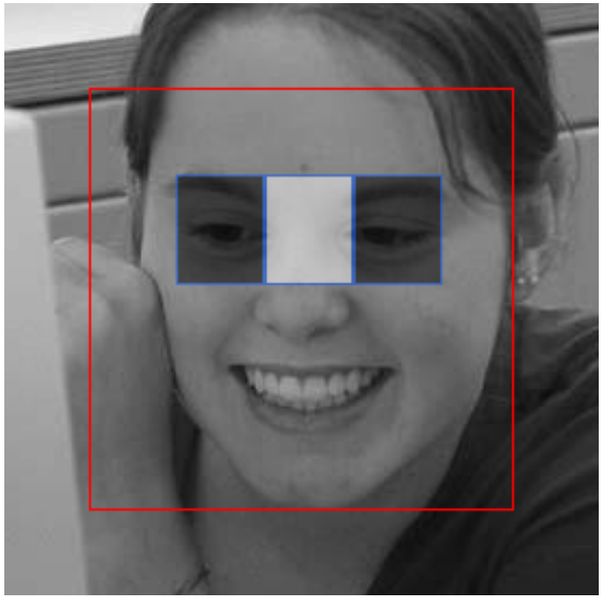
Silný (přesný) klasifikátor

$$H(x;\boldsymbol{\alpha}) = \begin{cases} +1 & \text{pro} \quad f(x;\boldsymbol{\alpha}) \geq 0 \\ -1 & \text{pro} \quad f(x;\boldsymbol{\alpha}) < 0 \end{cases} \quad \text{kde} \quad f(x;\boldsymbol{\alpha}) = \alpha_1 h_1(x) + \alpha_2 h_2(x) + \ldots + \alpha_m h_m(x)$$

je složen ze slabých, ale rychlých klasifikátorů

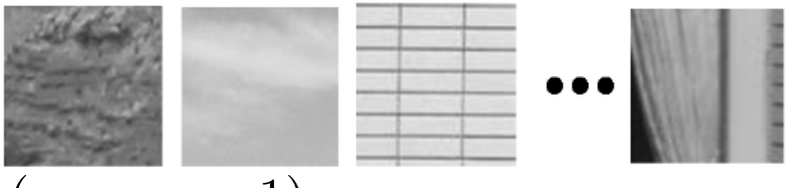
$$h_i(x) = \operatorname{sign}\left(\sum_{(u,v)\in A_i^+(x)} x(u,v) - \sum_{(u,v)\in A_i^-(x)} x(u,v) + \theta \right)$$





Problém učení: $\min_{\boldsymbol{\alpha}} \sum_{i=1}^{l} [\![H(x_i;\boldsymbol{\alpha}) \neq y_i]\!]$ za podmínky, že počet nenulových elementů ve vektoru všech vah $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ je právě n

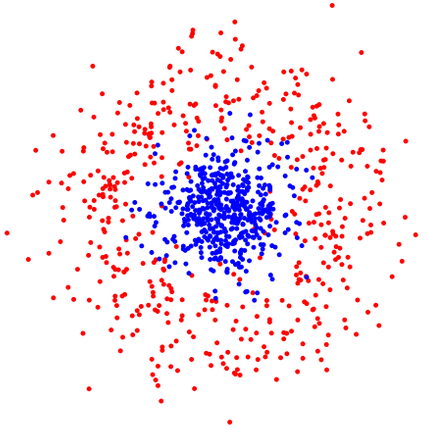
$(x_i, y_i = +1)$



$(x_i, y_i = -1)$

AdaBoost algoritmus

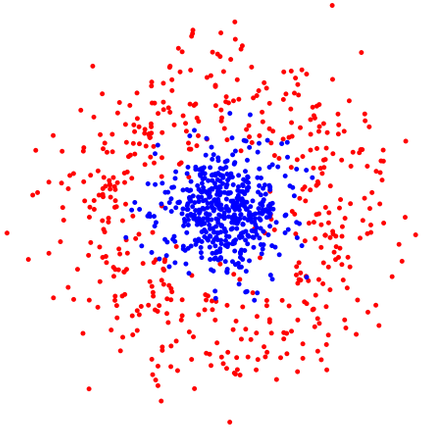
Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$



AdaBoost algoritmus

Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$

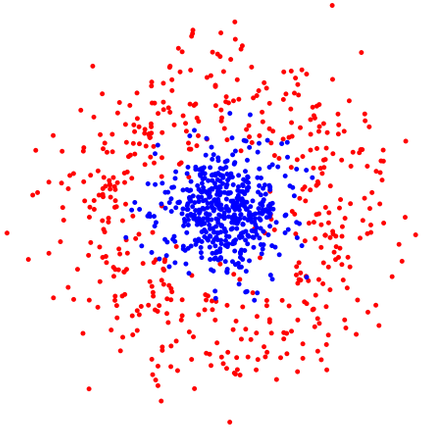


AdaBoost algoritmus

Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$

For $t = 1, ..., T$:



AdaBoost algoritmus

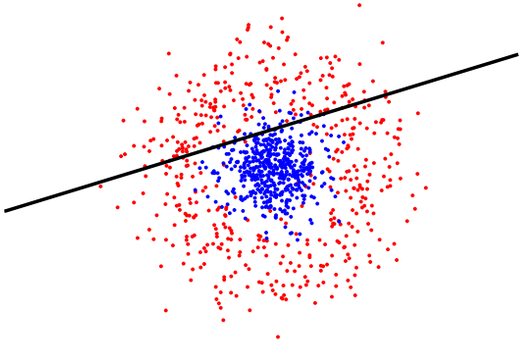
Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$

$t = 1$

For $t = 1, ..., T$:

- Find $h_t = \arg \min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i) [\![y_i \neq h_j(x_i)]\!]$



AdaBoost algoritmus

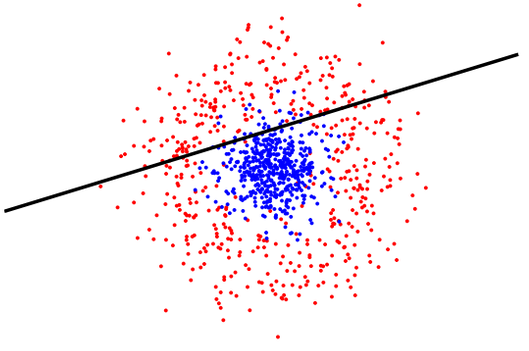
Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$

$t = 1$

For $t = 1, ..., T$:

- Find $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i)[\![y_i \neq h_j(x_i)]\!]$
- If $\epsilon_t \geq 1/2$ then stop



AdaBoost algoritmus

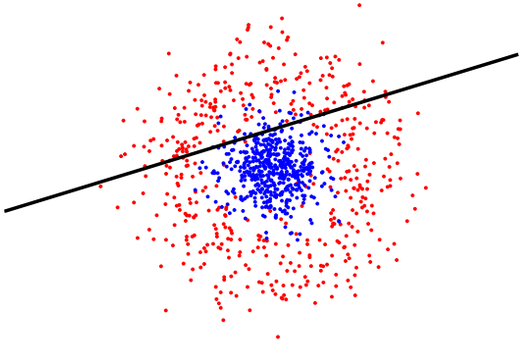
Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$

For $t = 1, ..., T$:

- Find $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i)[\![y_i \neq h_j(x_i)]\!]$
- If $\epsilon_t \geq 1/2$ then stop
- Set $\alpha_t = \frac{1}{2}\log(\frac{1-\epsilon_t}{\epsilon_t})$

$t = 1$



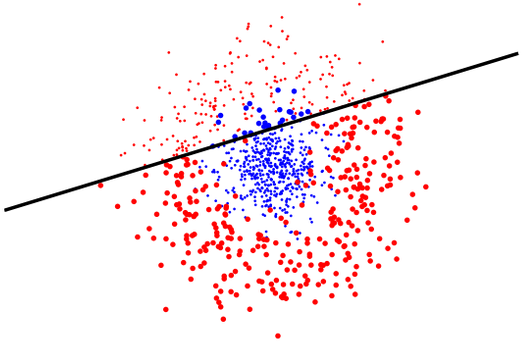
AdaBoost algoritmus

Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$

$t = 1$

For $t = 1, ..., T$:



- Find $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i)[\![y_i \neq h_j(x_i)]\!]$
- If $\epsilon_t \geq 1/2$ then stop
- Set $\alpha_t = \frac{1}{2}\log(\frac{1-\epsilon_t}{\epsilon_t})$
- Update

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is normalisation factor

AdaBoost algoritmus

Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$

For $t = 1, ..., T$:

- Find $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i)[\![y_i \neq h_j(x_i)]\!]$
- If $\epsilon_t \geq 1/2$ then stop
- Set $\alpha_t = \frac{1}{2}\log(\frac{1-\epsilon_t}{\epsilon_t})$
- Update

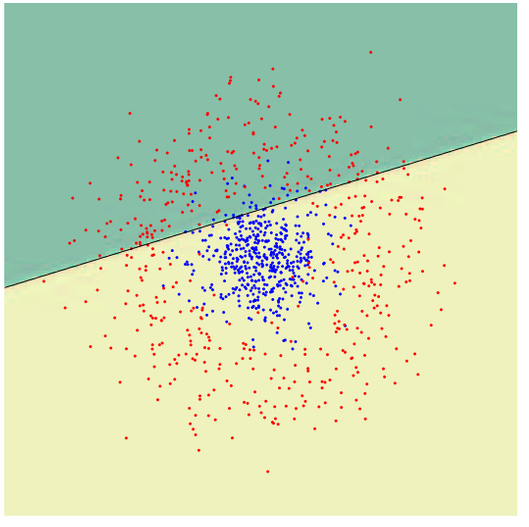
$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

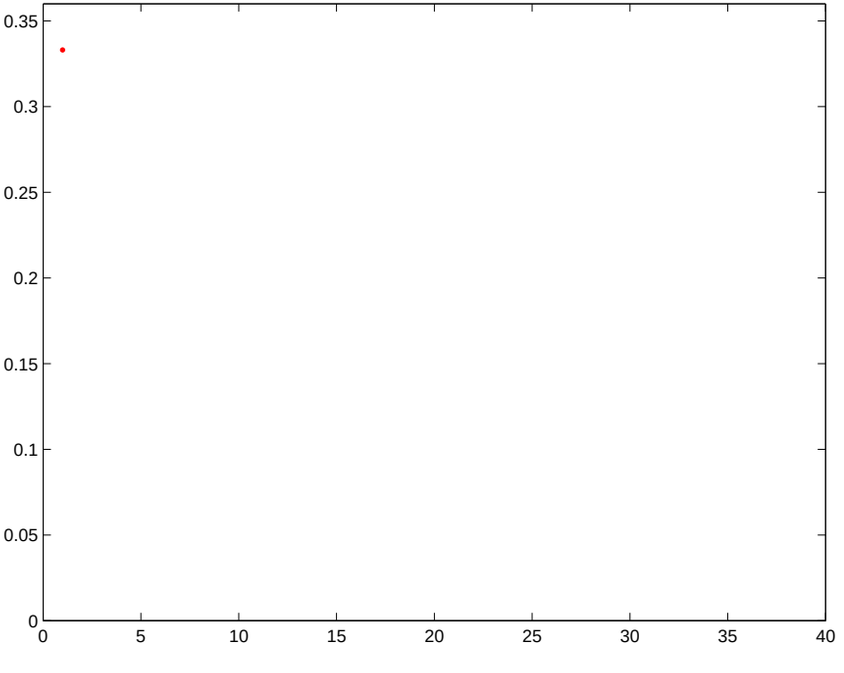
where Z_t is normalisation factor

Output the final classifier:

$$H(x) = \text{sign}\left(\alpha_1 h_1(x)\right)$$

$t = 1$





AdaBoost algoritmus

Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$

For $t = 1, ..., T$:

- Find $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i) [\![y_i \neq h_j(x_i)]\!]$
- If $\epsilon_t \geq 1/2$ then stop
- Set $\alpha_t = \frac{1}{2} \log(\frac{1-\epsilon_t}{\epsilon_t})$
- Update

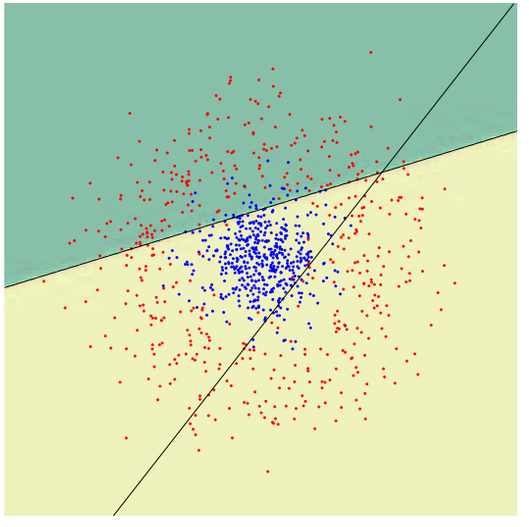
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

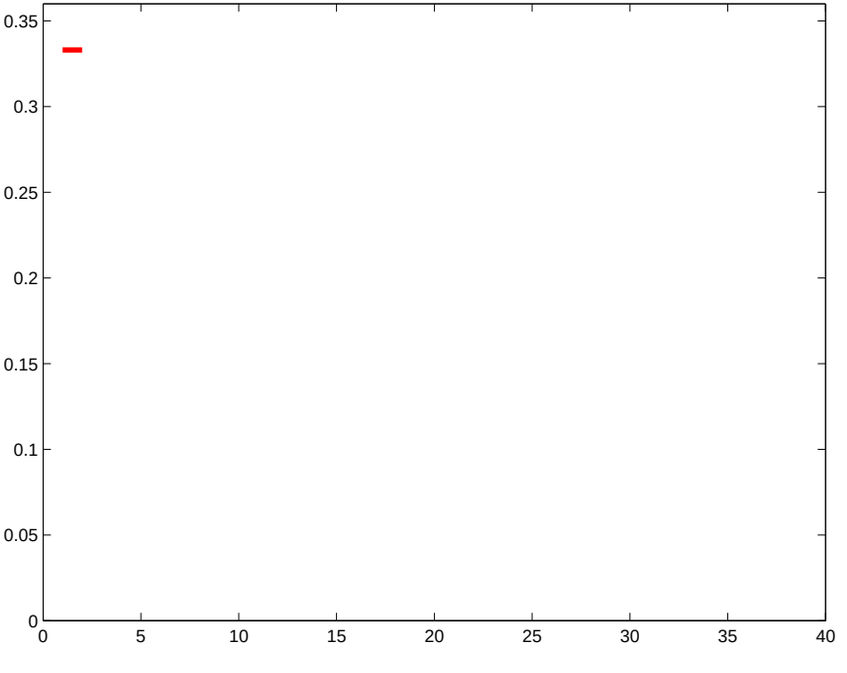
where Z_t is normalisation factor

Output the final classifier:

$$H(x) = \text{sign}\left(\alpha_1 h_1(x) + \alpha_2 h_2(x)\right)$$

$t = 2$





AdaBoost algoritmus

Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$

For $t = 1, ..., T$:

- Find $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i) [\![y_i \neq h_j(x_i)]\!]$
- If $\epsilon_t \geq 1/2$ then stop
- Set $\alpha_t = \frac{1}{2} \log(\frac{1-\epsilon_t}{\epsilon_t})$
- Update

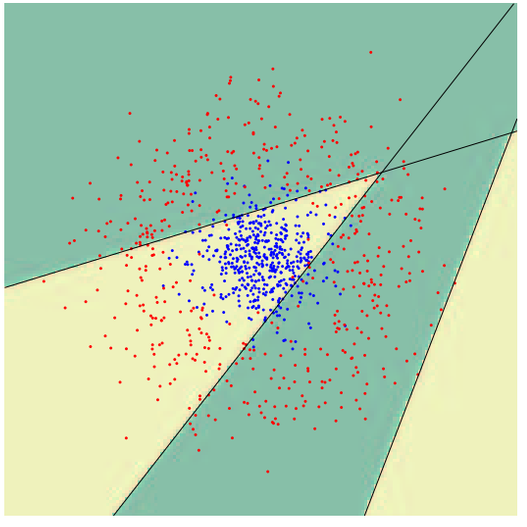
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

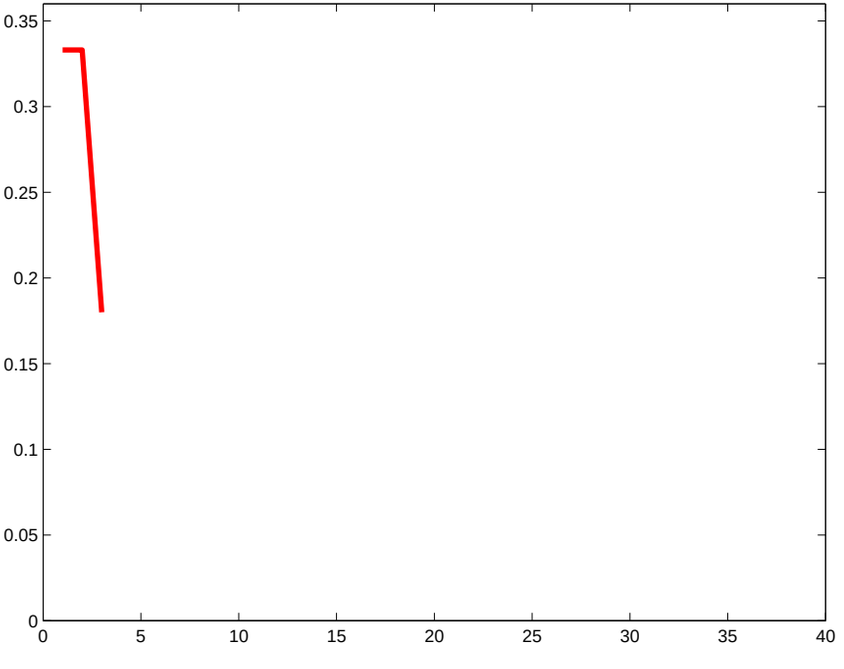
where Z_t is normalisation factor

Output the final classifier:

$$H(x) = \text{sign}\left(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x)\right)$$

$t = 3$





AdaBoost algoritmus

Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$

For $t = 1, ..., T$:

- Find $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i) [\![y_i \neq h_j(x_i)]\!]$
- If $\epsilon_t \geq 1/2$ then stop
- Set $\alpha_t = \frac{1}{2} \log(\frac{1-\epsilon_t}{\epsilon_t})$
- Update

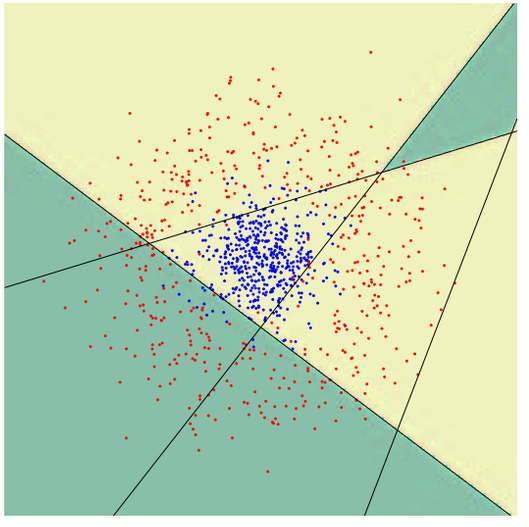
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

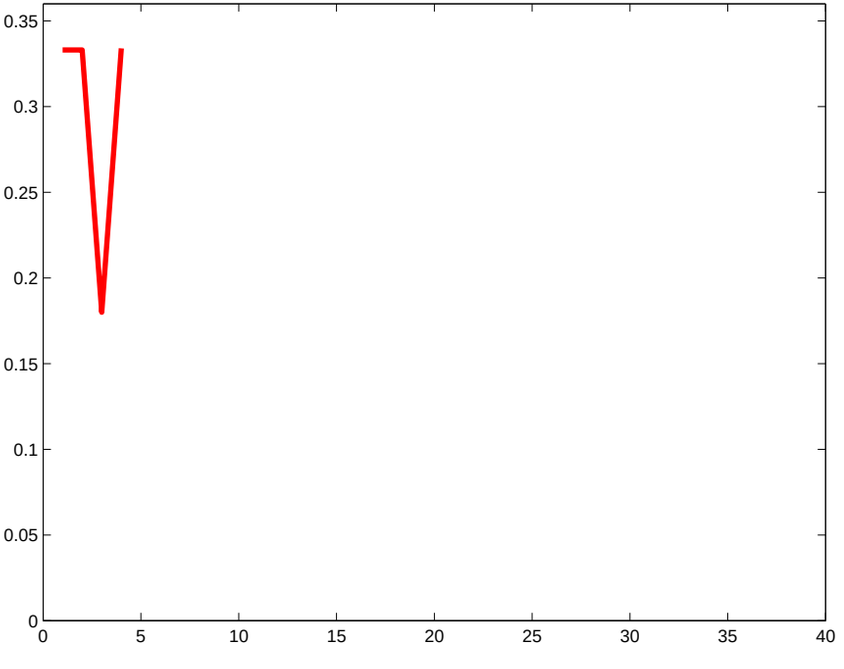
where Z_t is normalisation factor

Output the final classifier:

$$H(x) = \text{sign}\left(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \ldots + \alpha_4 h_4(x)\right)$$

$t = 4$





AdaBoost algoritmus

Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$
Initialise weights $D_1(i) = 1/m$
For $t = 1, ..., T$:

- Find $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i)[\![y_i \neq h_j(x_i)]\!]$
- If $\epsilon_t \geq 1/2$ then stop
- Set $\alpha_t = \frac{1}{2}\log(\frac{1-\epsilon_t}{\epsilon_t})$
- Update

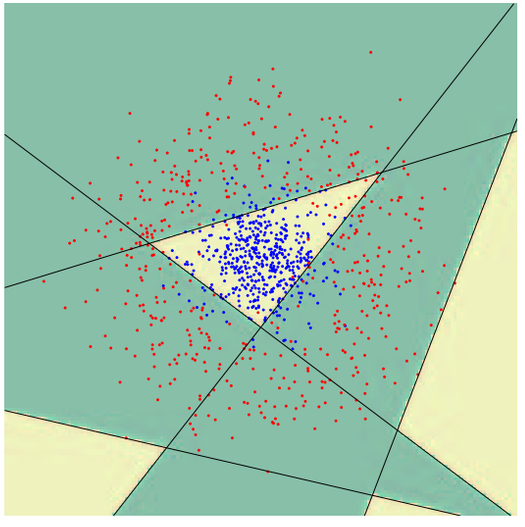
$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

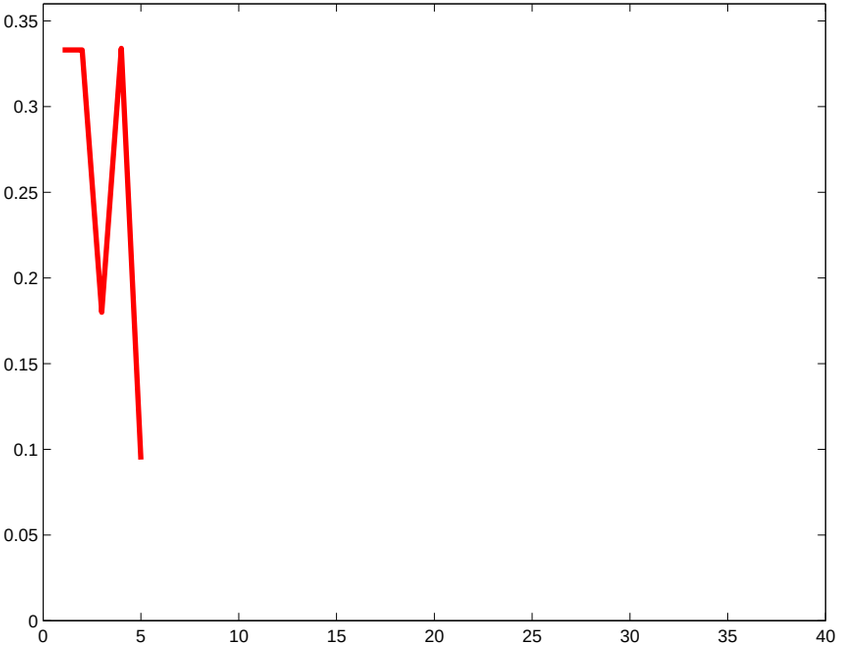
where Z_t is normalisation factor

Output the final classifier:

$$H(x) = \text{sign}\left(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \ldots + \alpha_5 h_5(x)\right)$$

$t = 5$





AdaBoost algoritmus

Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

Initialise weights $D_1(i) = 1/m$

For $t = 1, ..., T$:

- Find $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i)[\![y_i \neq h_j(x_i)]\!]$
- If $\epsilon_t \geq 1/2$ then stop
- Set $\alpha_t = \frac{1}{2}\log(\frac{1-\epsilon_t}{\epsilon_t})$
- Update

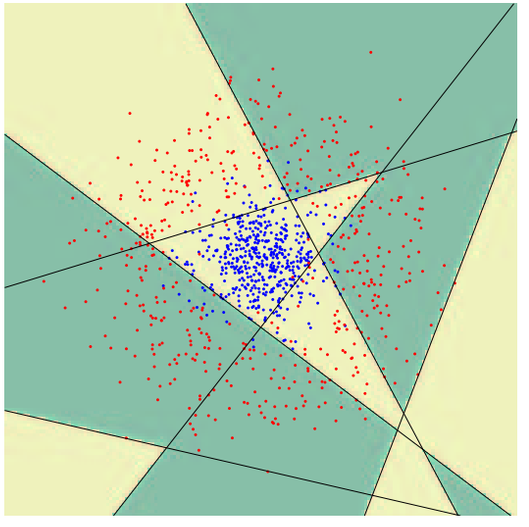
$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

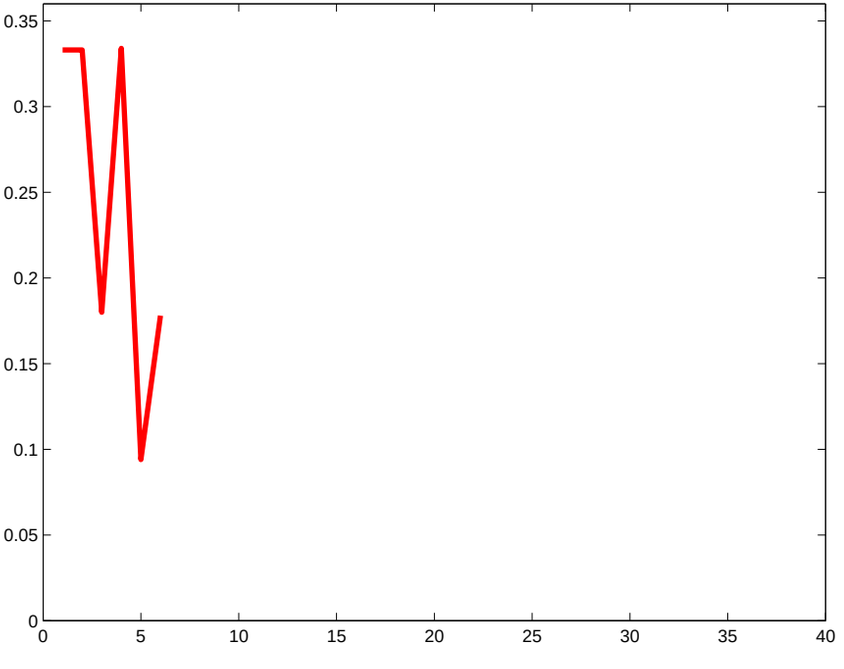
where Z_t is normalisation factor

Output the final classifier:

$$H(x) = \text{sign}\left(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \ldots + \alpha_6 h_6(x)\right)$$

$t = 6$





AdaBoost algoritmus

Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$
Initialise weights $D_1(i) = 1/m$
For $t = 1, ..., T$:

- Find $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i)[\![y_i \neq h_j(x_i)]\!]$
- If $\epsilon_t \geq 1/2$ then stop
- Set $\alpha_t = \frac{1}{2}\log(\frac{1-\epsilon_t}{\epsilon_t})$
- Update

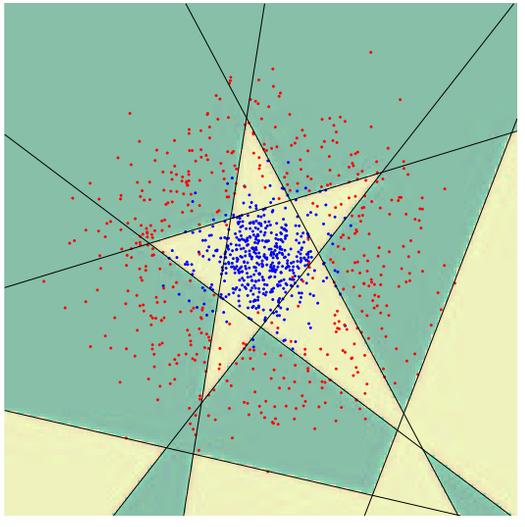
$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

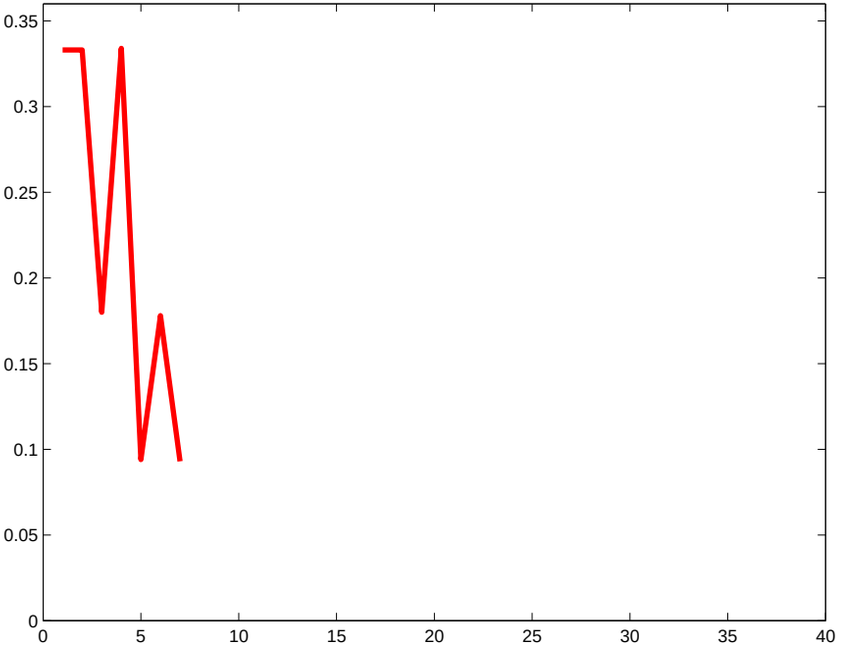
where Z_t is normalisation factor

Output the final classifier:

$$H(x) = \text{sign}\left(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \ldots + \alpha_7 h_7(x)\right)$$

$t = 7$





AdaBoost algoritmus

Given: $(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$
Initialise weights $D_1(i) = 1/m$
For $t = 1, ..., T$:

- Find $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i)[\![y_i \neq h_j(x_i)]\!]$
- If $\epsilon_t \geq 1/2$ then stop
- Set $\alpha_t = \frac{1}{2}\log(\frac{1-\epsilon_t}{\epsilon_t})$
- Update

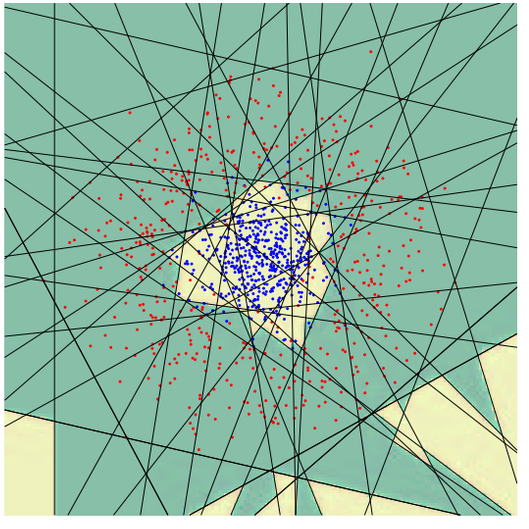
$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

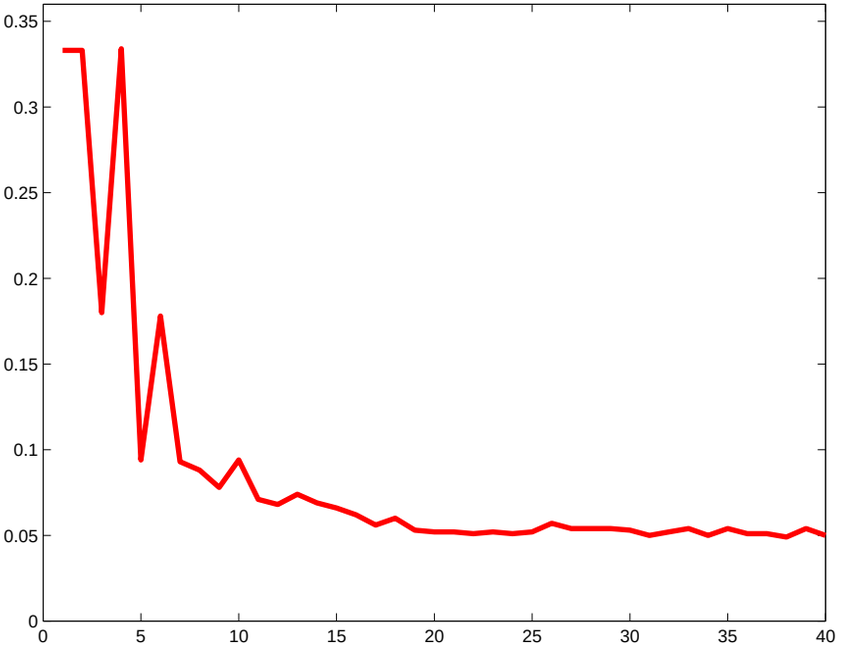
where Z_t is normalisation factor

Output the final classifier:

$$H(x) = \text{sign}\left(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \ldots + \alpha_{40} h_{40}(x)\right)$$

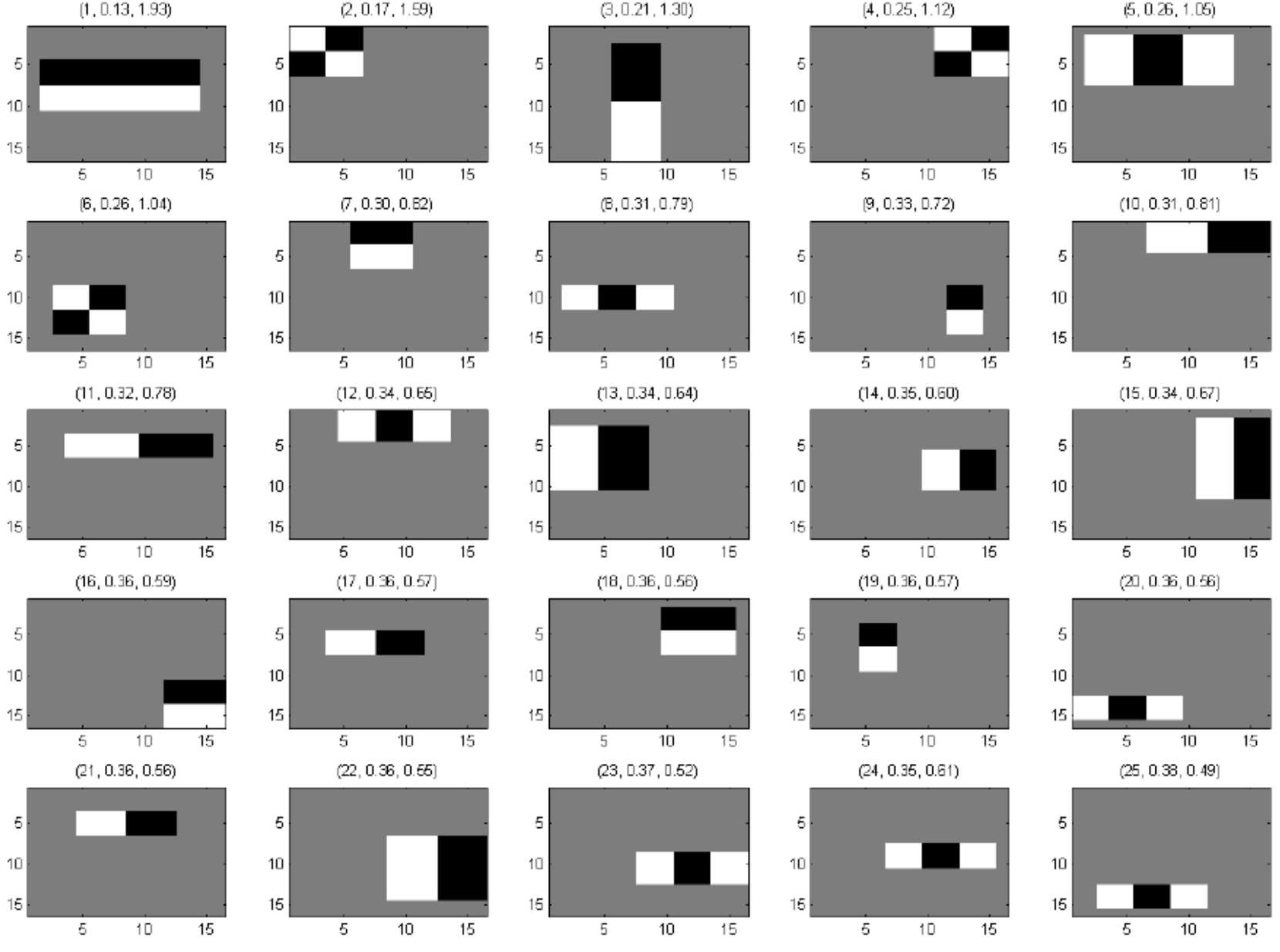
$t = 40$





Příklady nalezených slabých klasifikátorů pomocí AdaBoostu

Po prvních 25 iteracích.

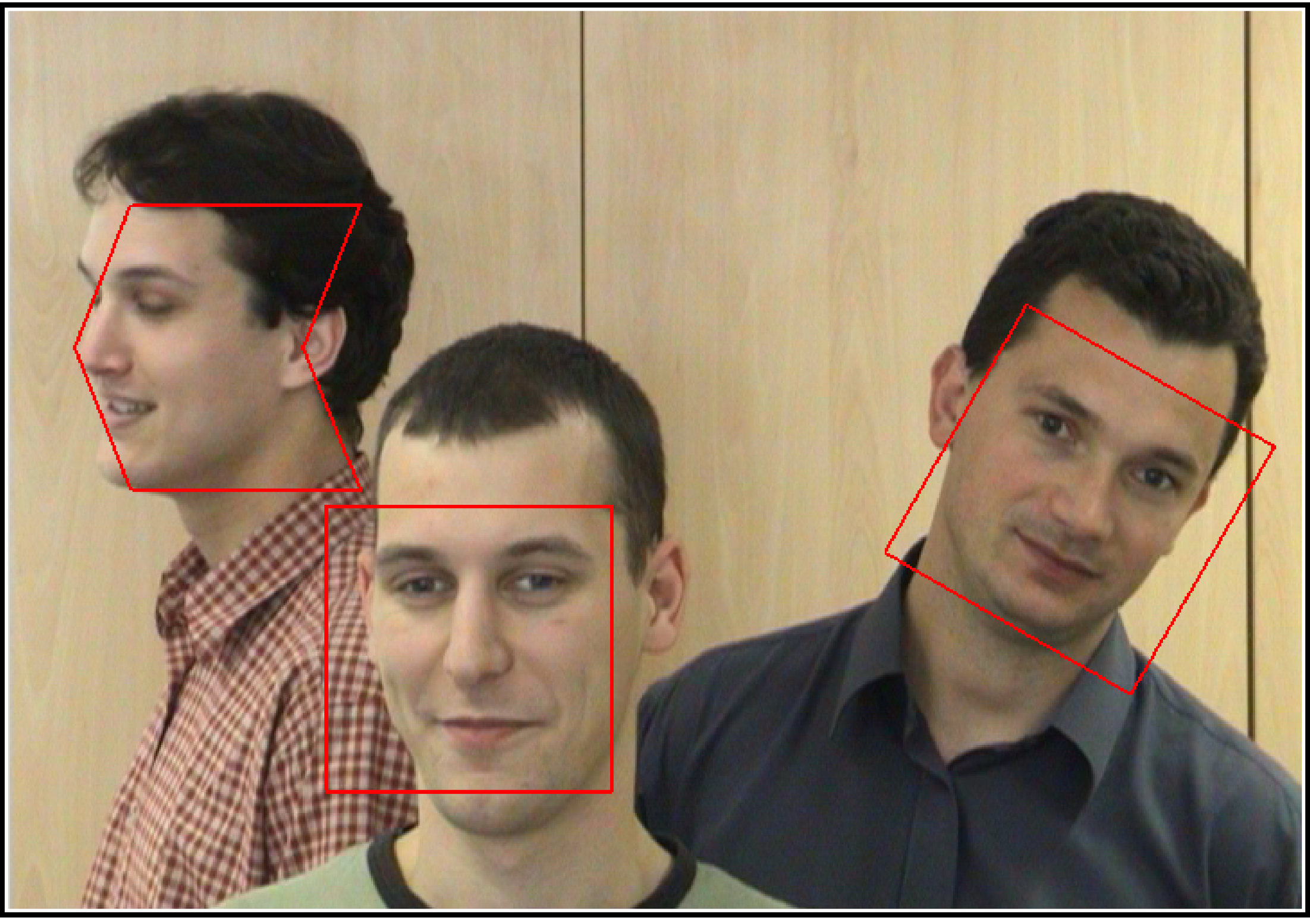


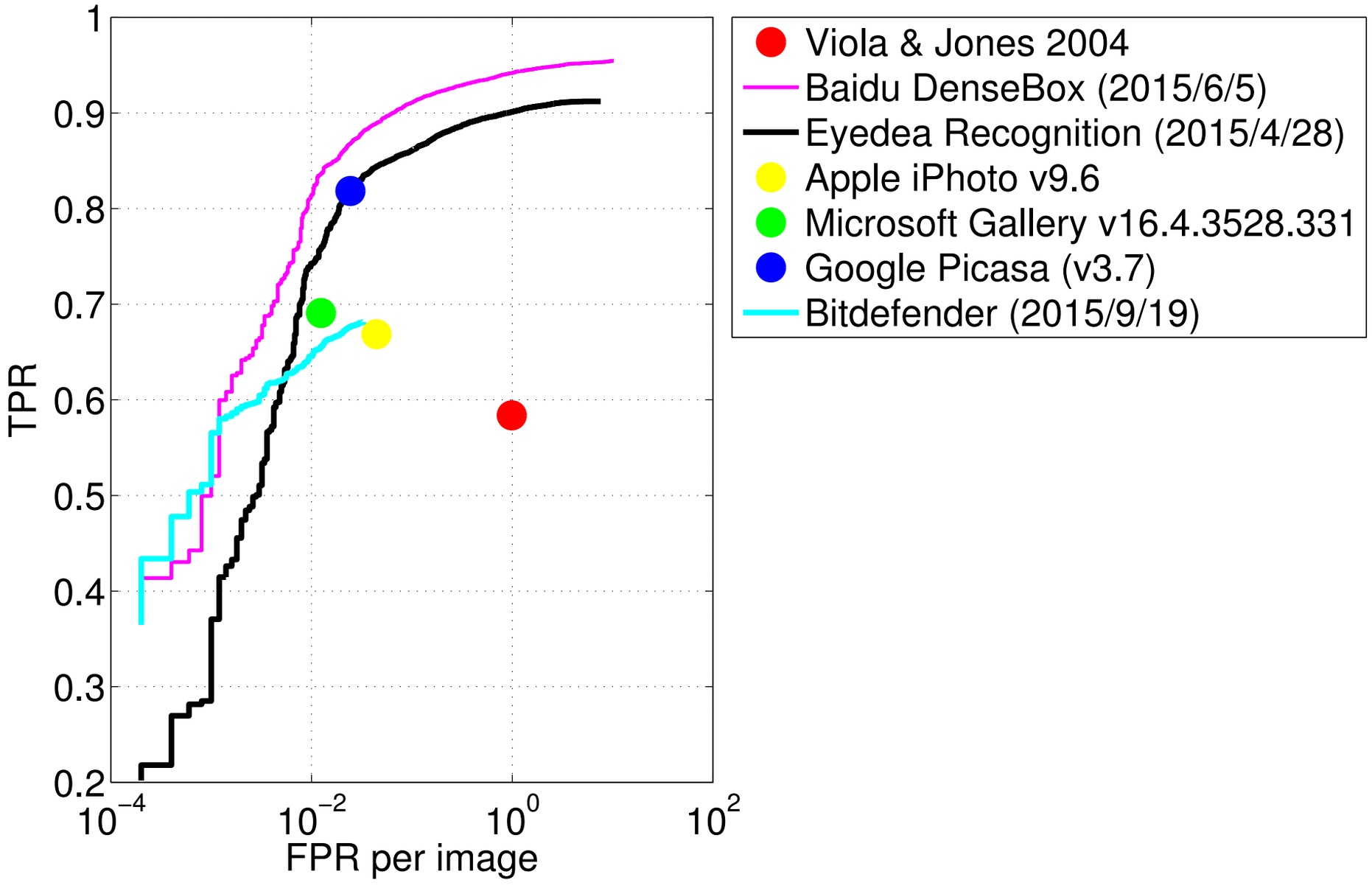
Homework: Take part in age estimation experiment

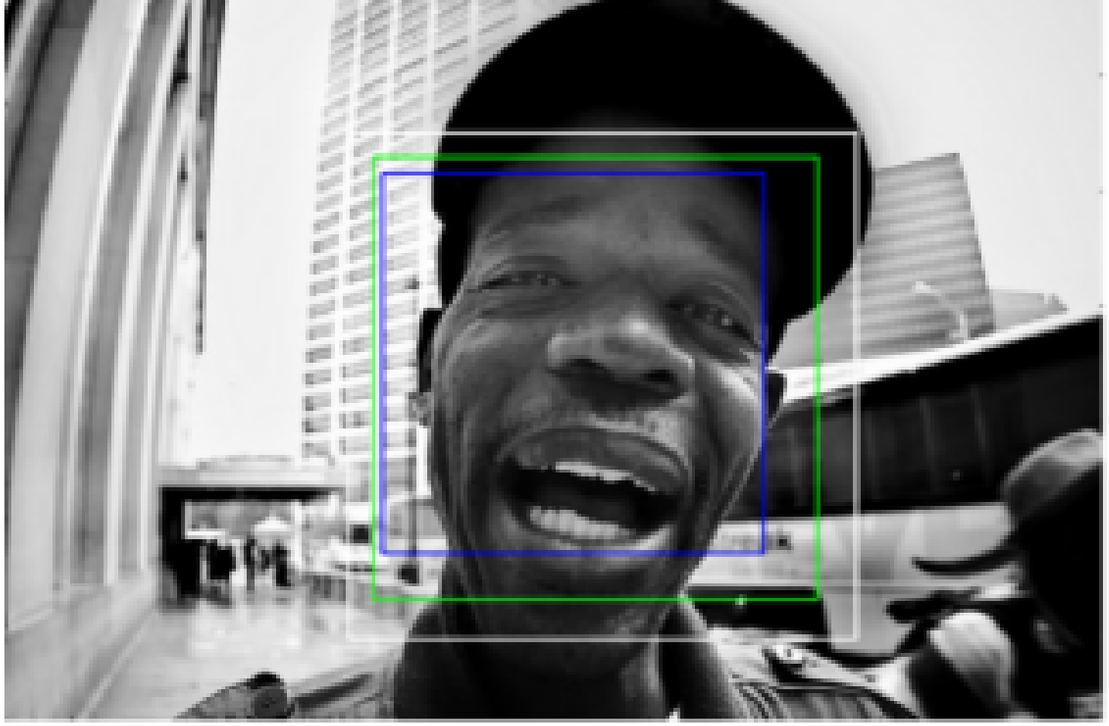
- Odhad věku: http://cmp.felk.cvut.cz/~xfrancv/ageannot/imdb/

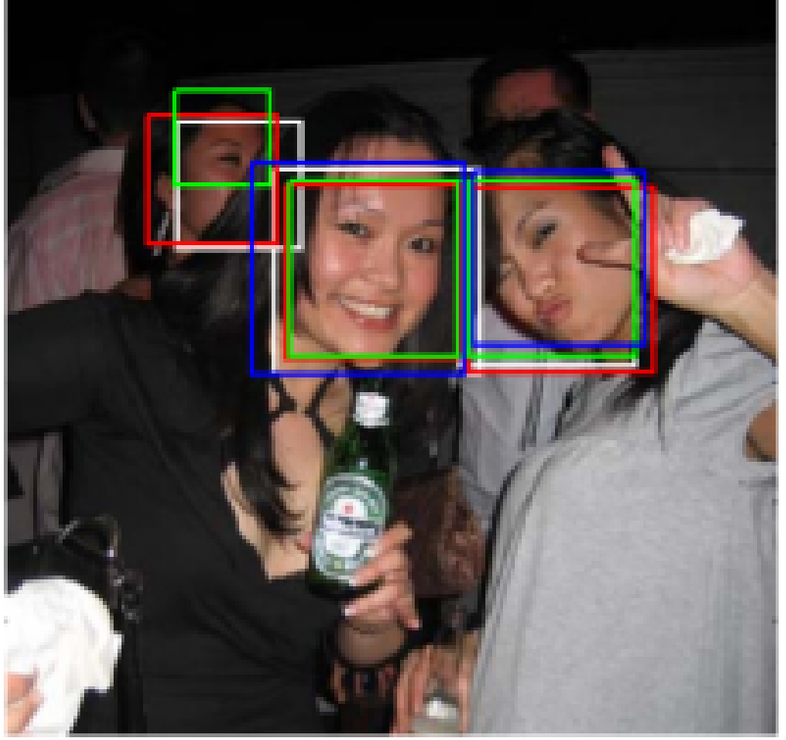
Konec

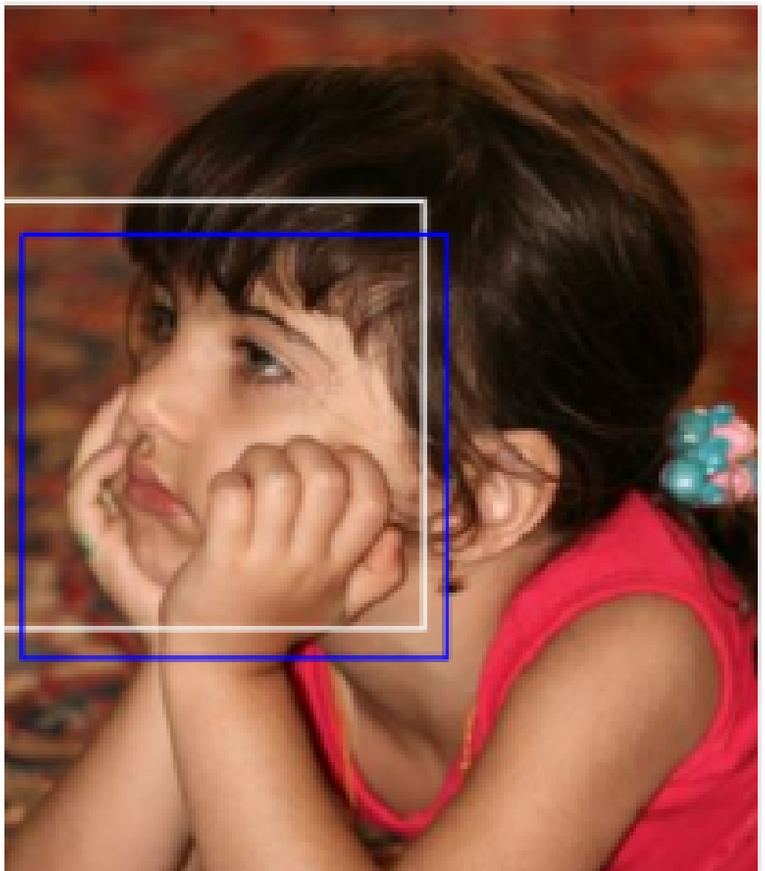

m p

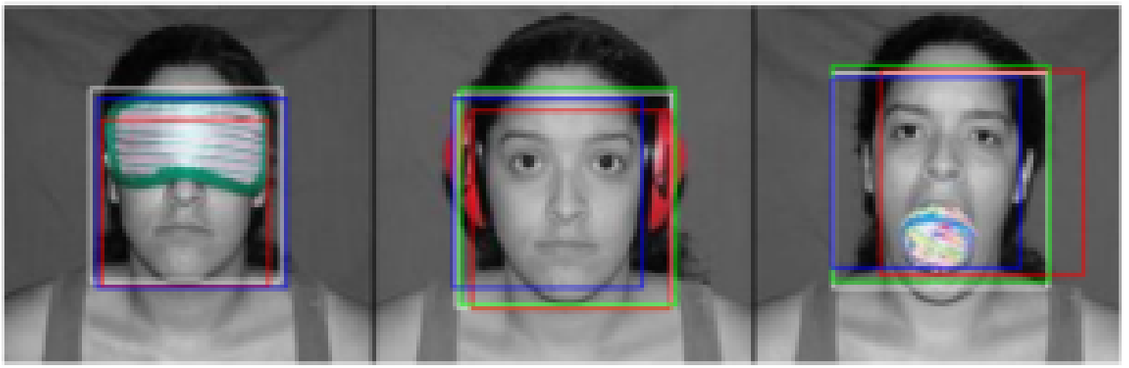


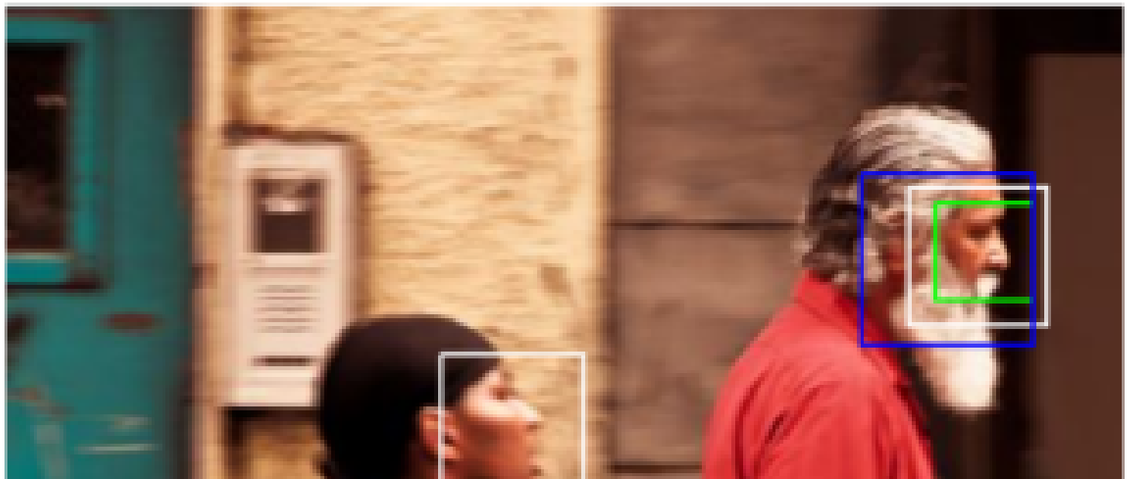

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
TPR
10^{-4}
10^{-2}
10^{0}
10^{2}
FPR per image
Viola & Jones 2004
Baidu DenseBox (2015/6/5)
Eyedea Recognition (2015/4/28)
Apple iPhoto v9.6
Microsoft Gallery v16.4.3528.331
Google Picasa (v3.7)
Bitdefender (2015/9/19)











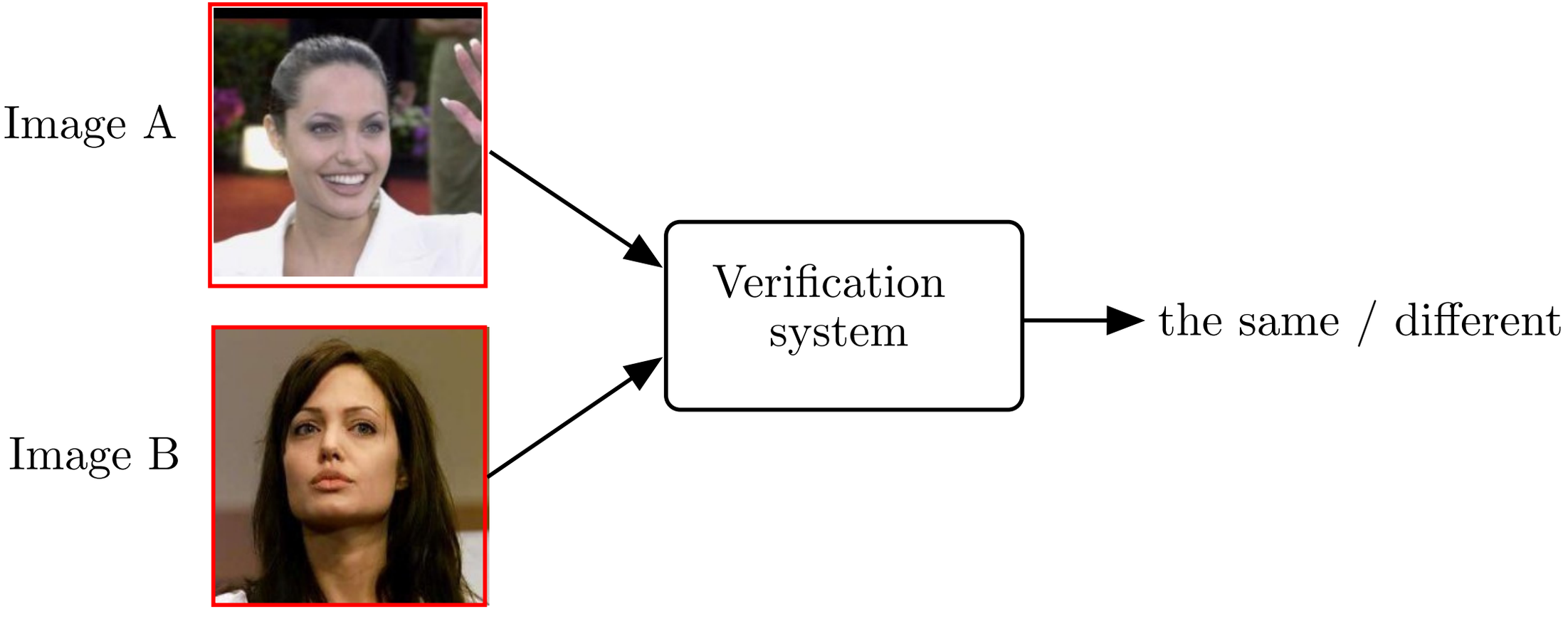
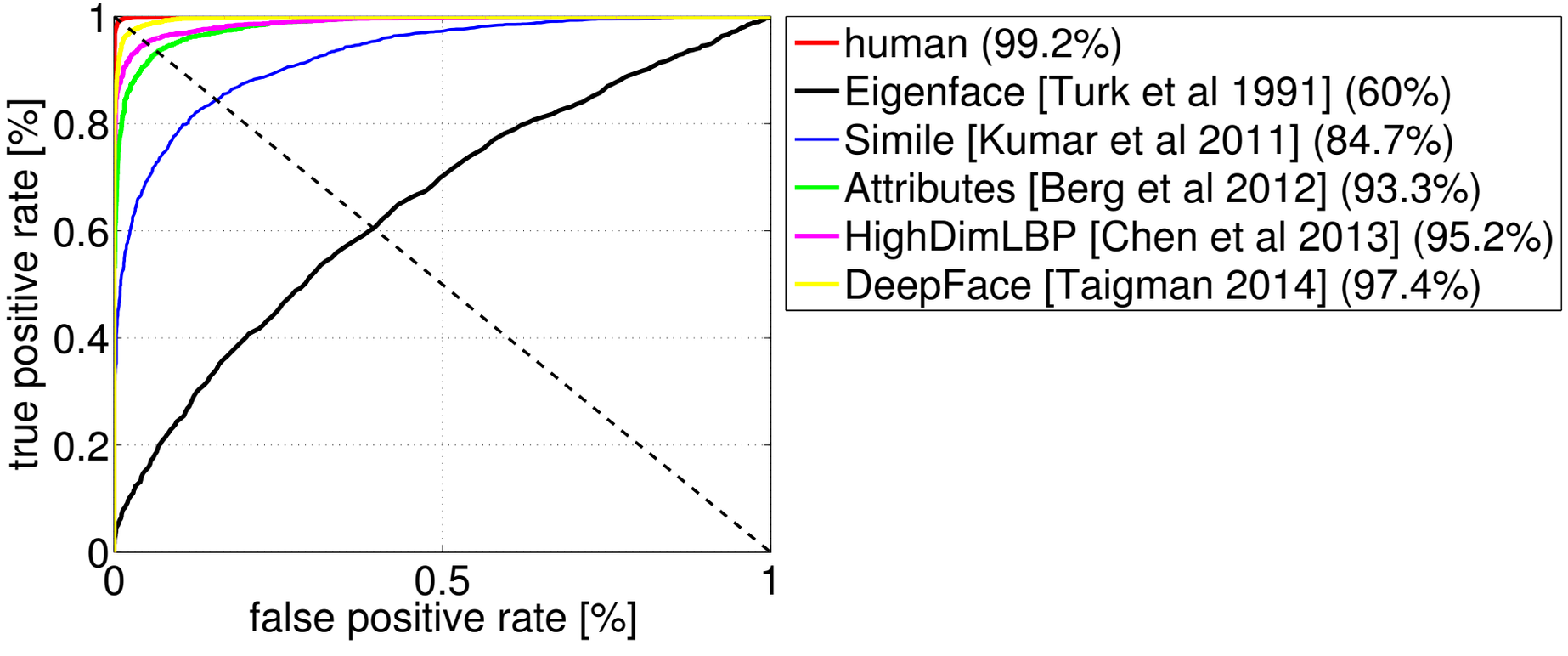
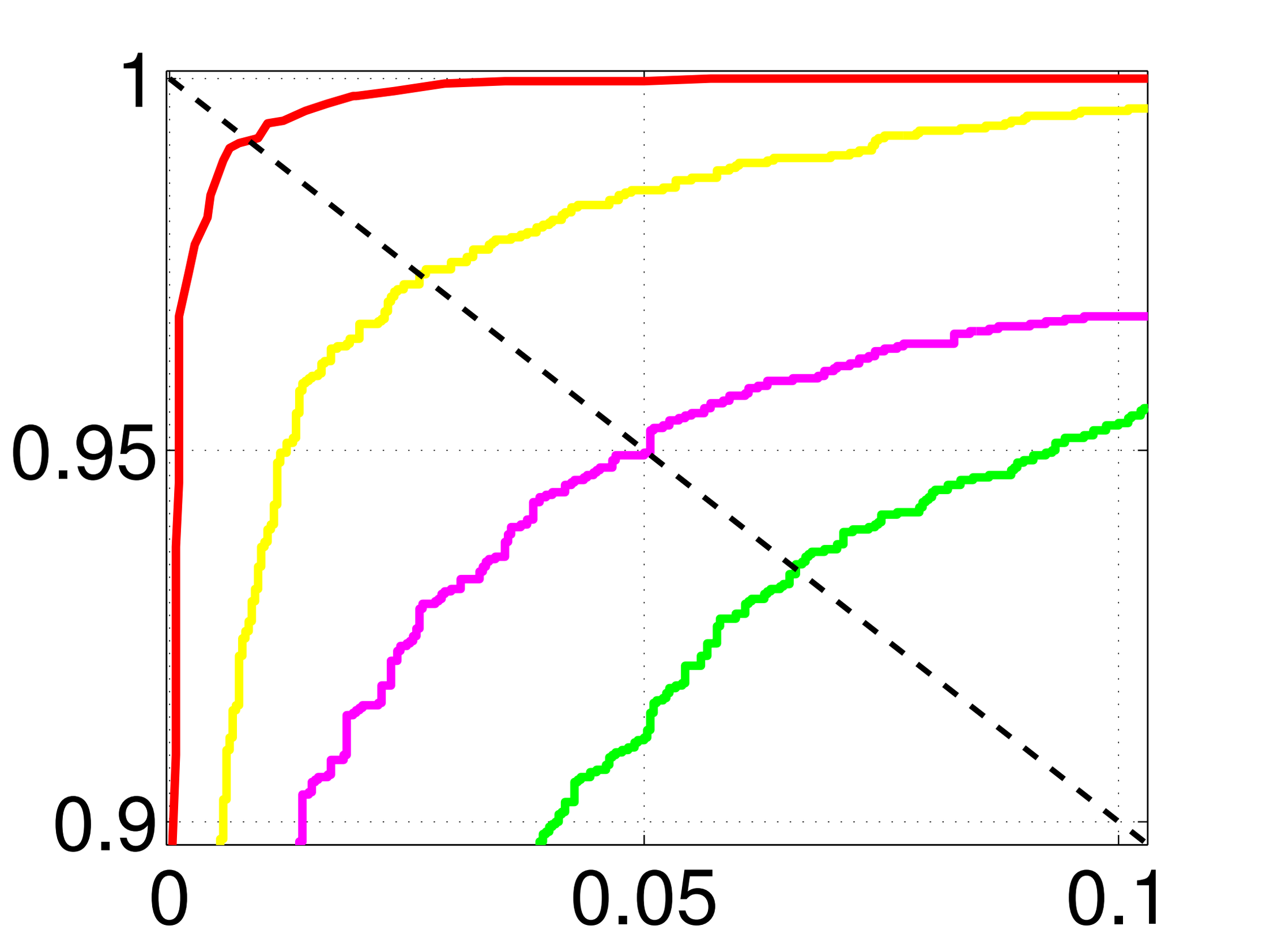
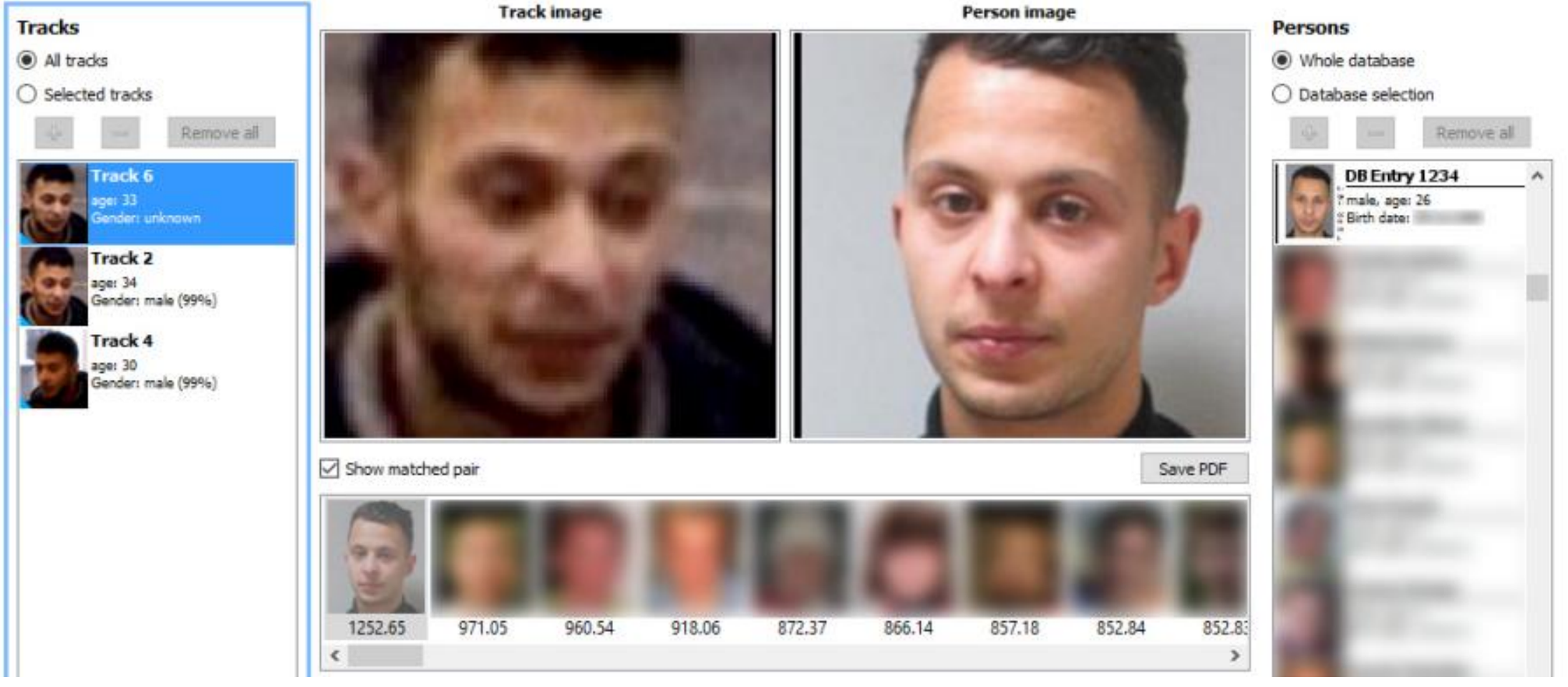
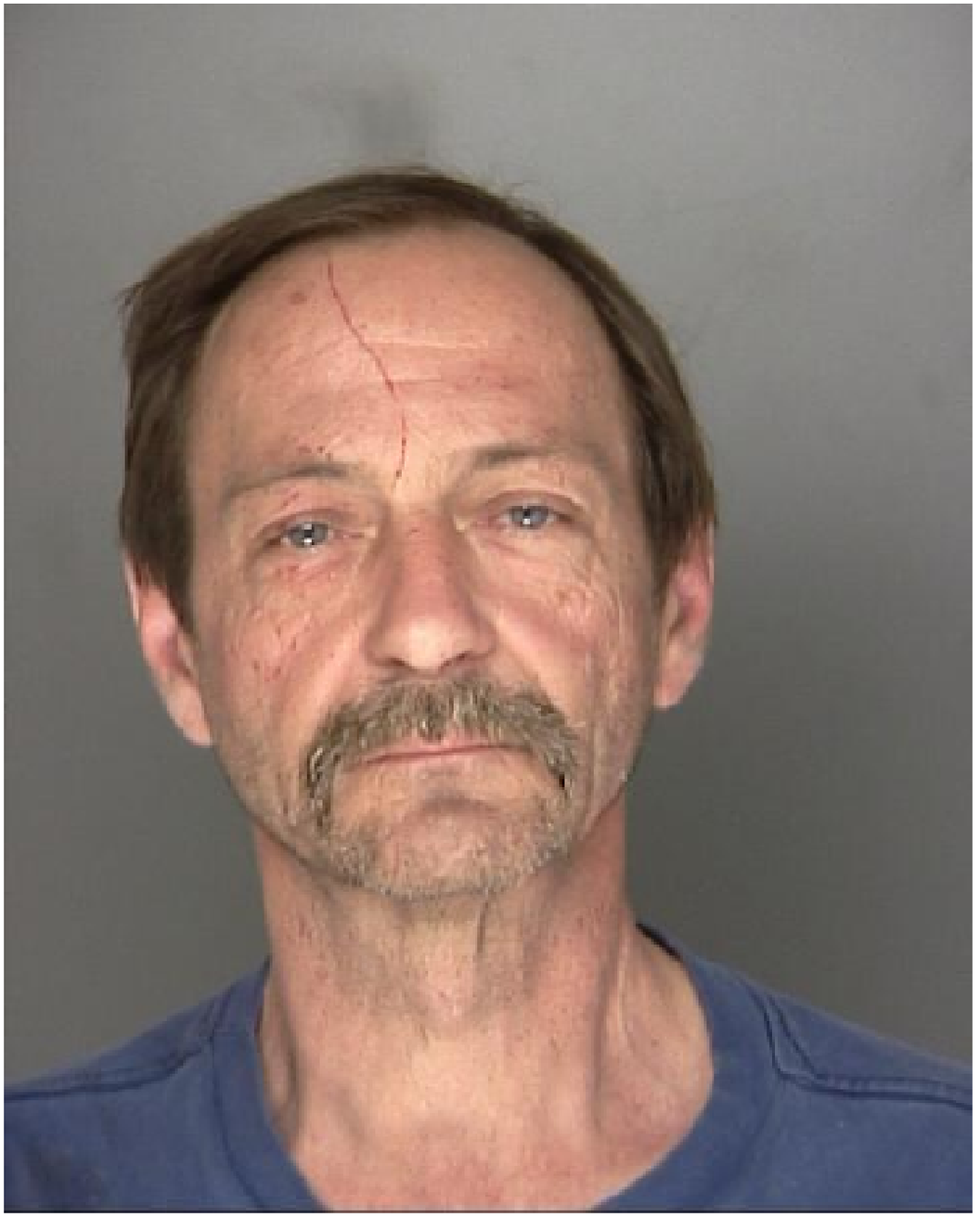


Image A
Image B
Verification system
the same / different


true positive rate [%]
false positive rate [%]
0
0.2
0.4
0.6
0.8
1
0
0.5
1
human (99.2%)
Eigenface [Turk et al 1991] (60%)
Simile [Kumar et al 2011] (84.7%)
Attributes [Berg et al 2012] (93.3%)
HighDimLBP [Chen et al 2013] (95.2%)
DeepFace [Taigman 2014] (97.4%)


1
0.95
0.9
0
0.05
0.1


Tracks
All tracks
Selected tracks
Remove all
Track 6
age: 33
Gender: unknown
Track 2
age: 34
Gender: male (99%)
Track 4
age: 30
Gender: male (99%)
Track image
Person image
Persons
Whole database
Database selection
Remove all
DB Entry 1234
male, age: 26
Birth date:
Show matched pair
Save PDF
1252.65
971.05
960.54
918.06
872.37
866.14
857.18
852.84
852.83



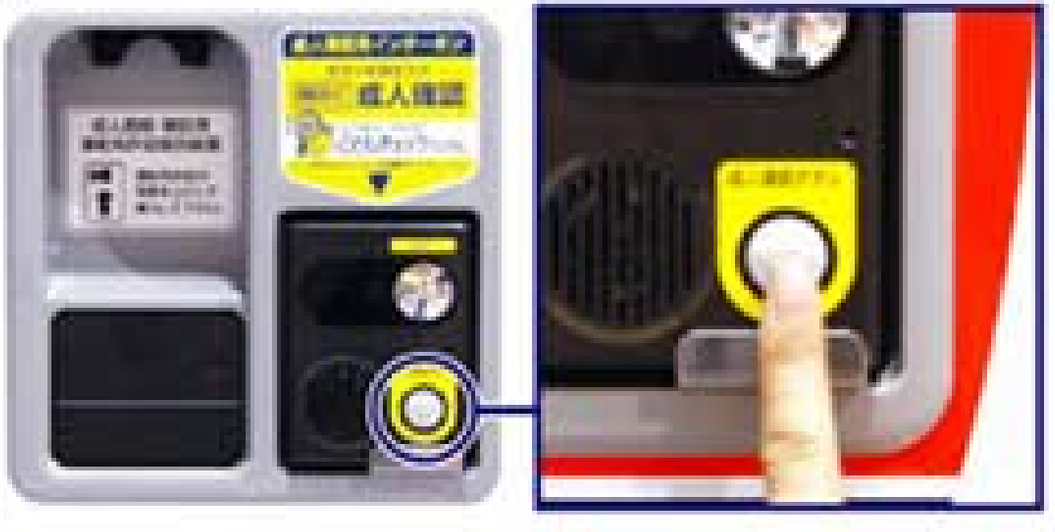
(a) Good quality mugshot

(b) Poor quality webcam

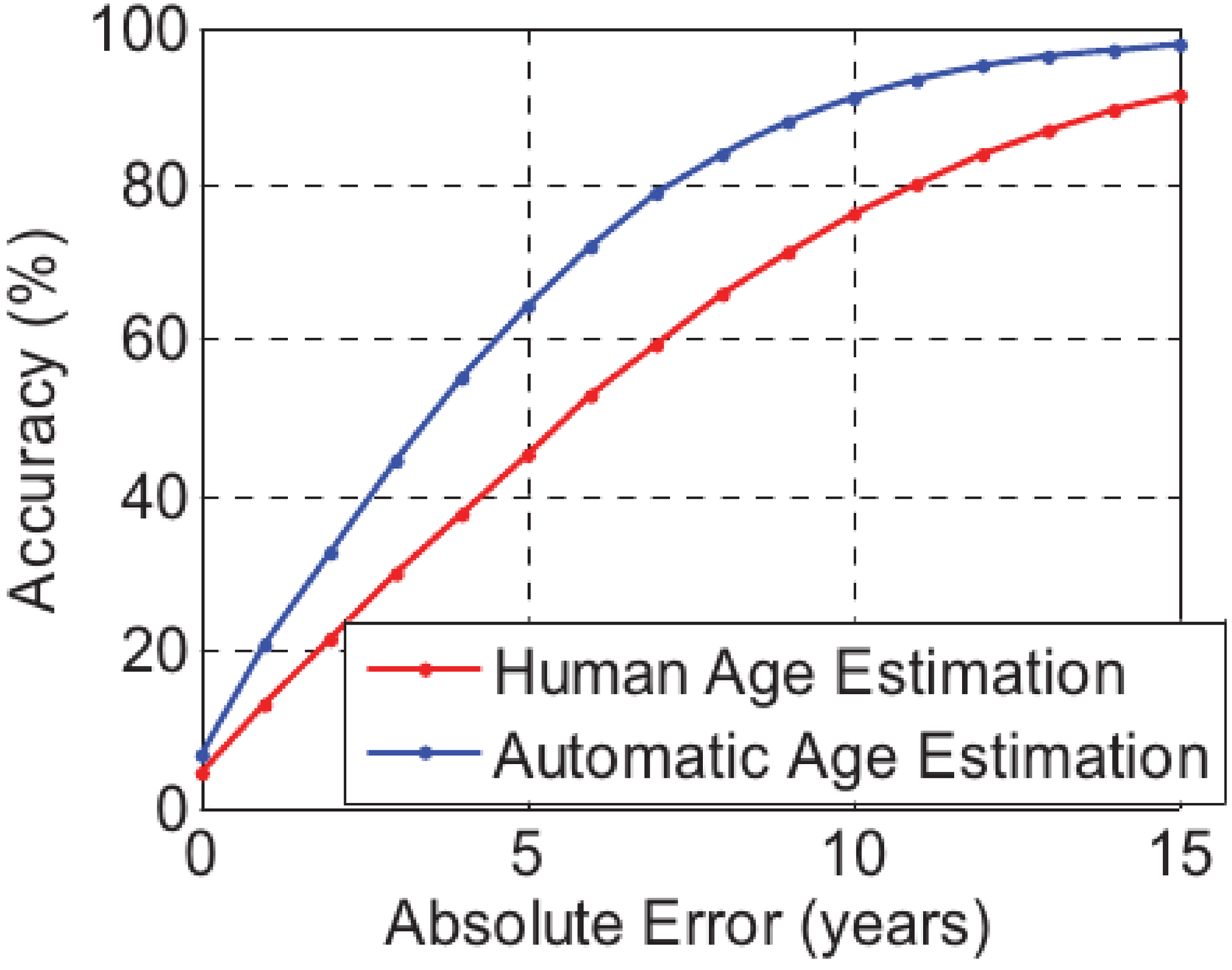
1.

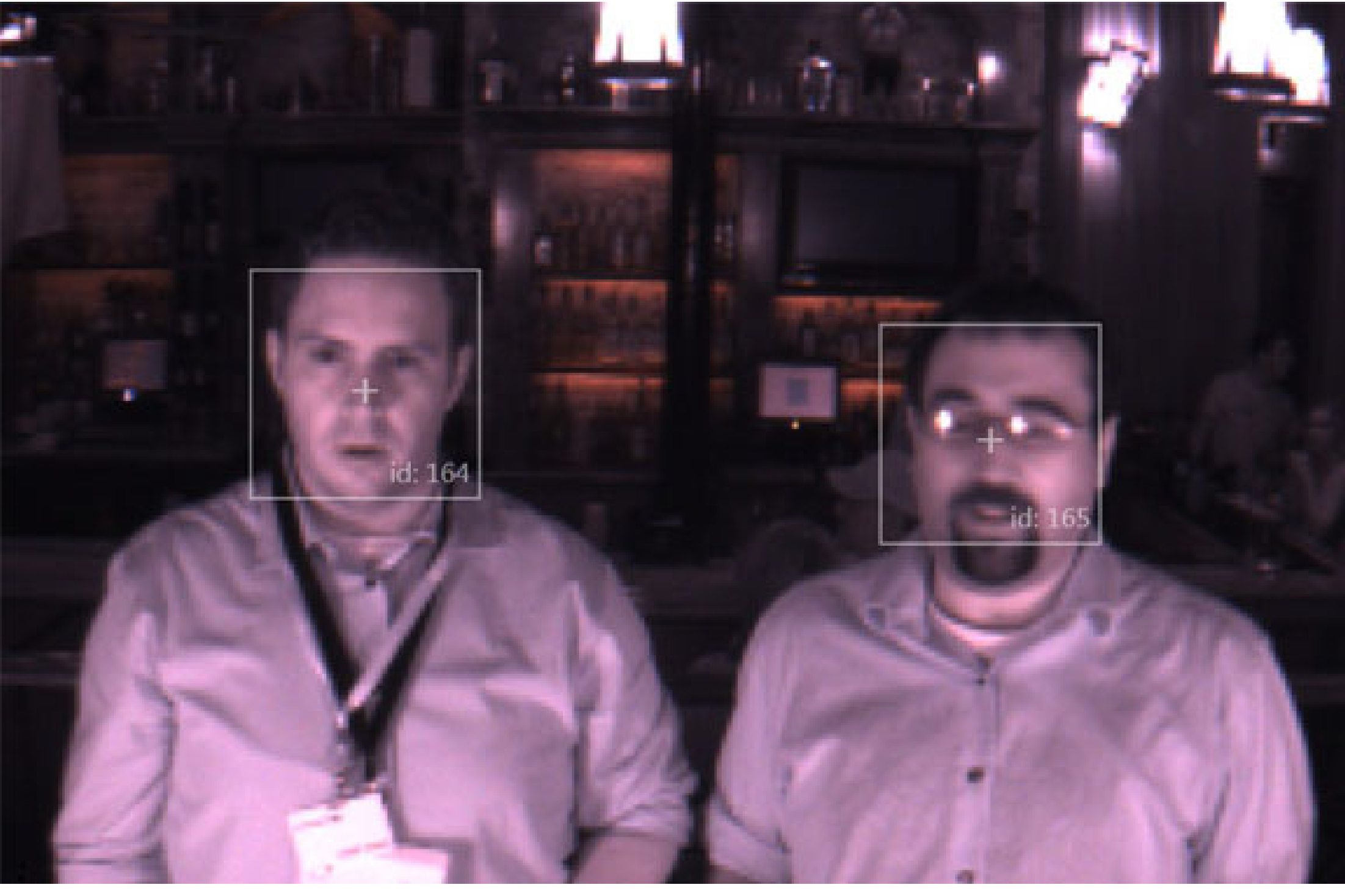
顔認証たばこ自販機右上部のインターホンのミラーにご注目。そのミラーに顔が映るようにお立ち頂きます。

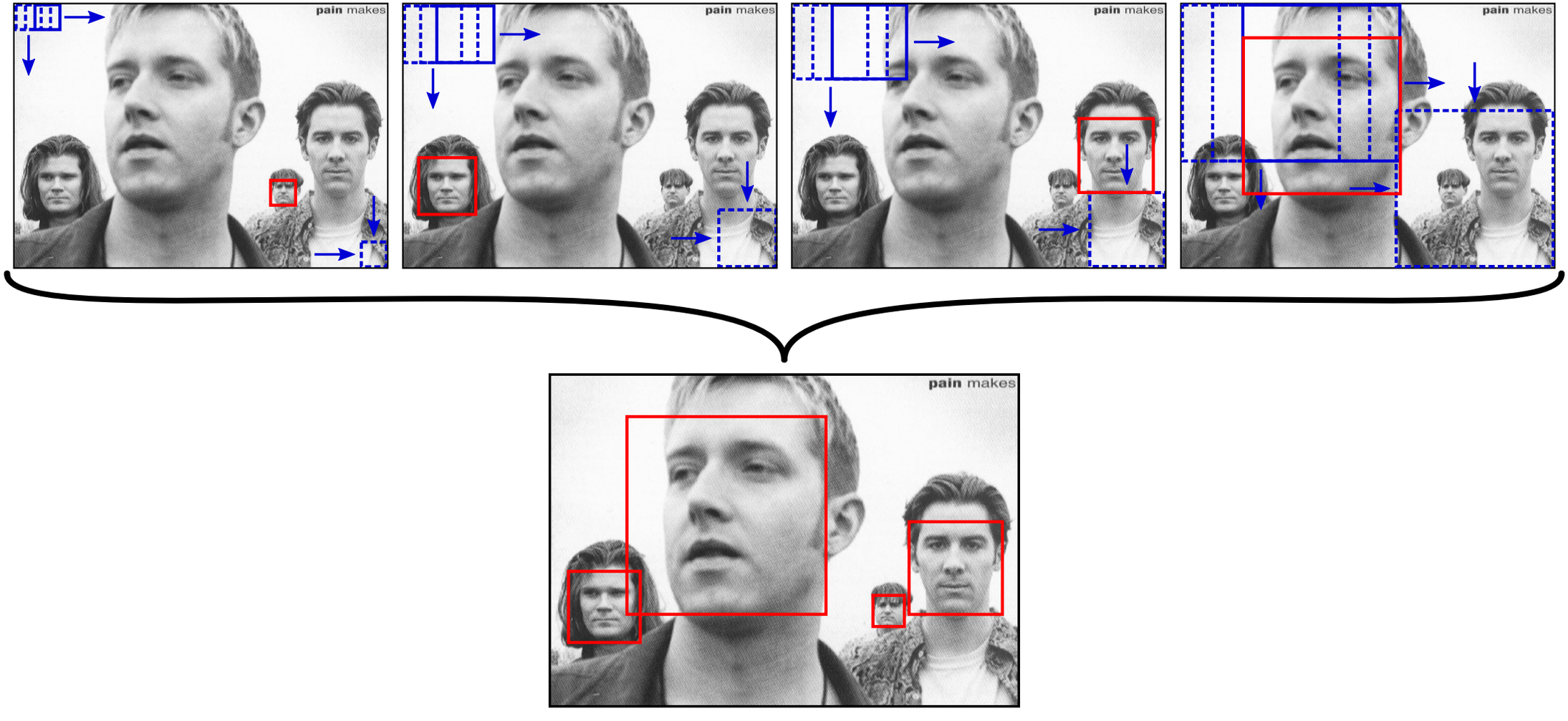
2.



3.

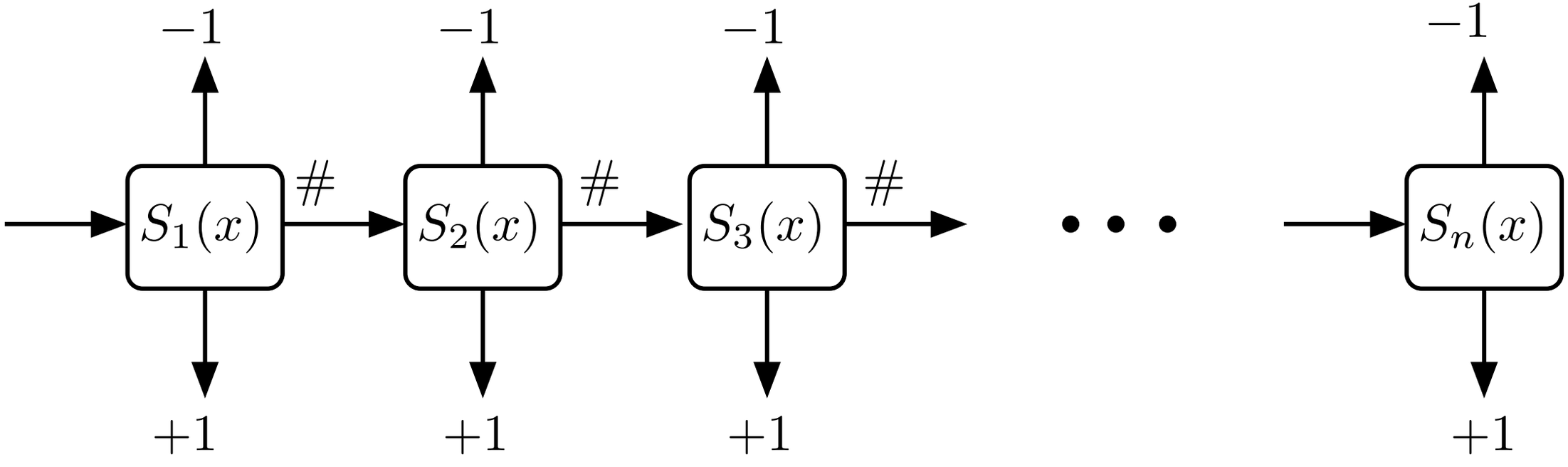

100
80
60
40
20
0
Accuracy (%)
0
5
10
15
Absolute Error (years)
Human Age Estimation
Automatic Age Estimation

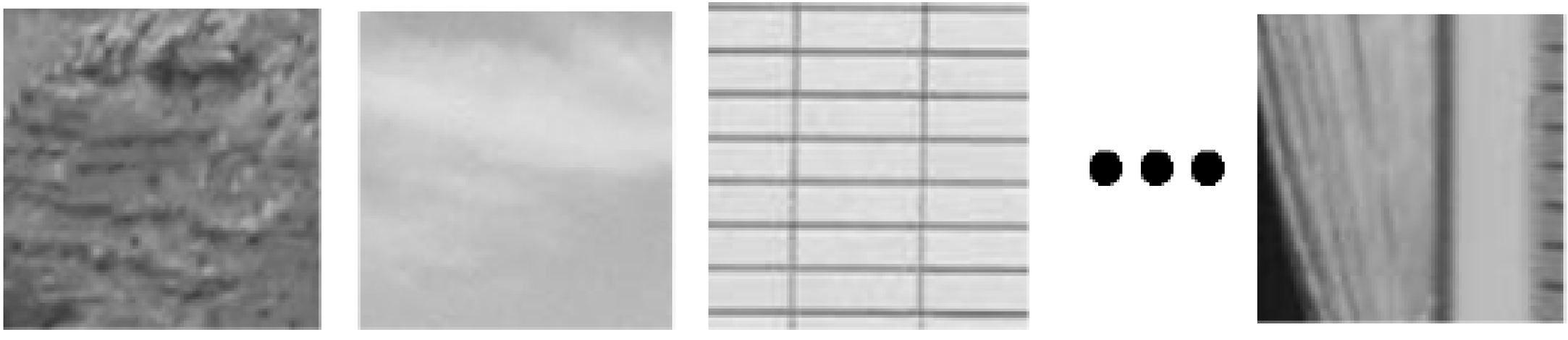

id: 164
id: 165

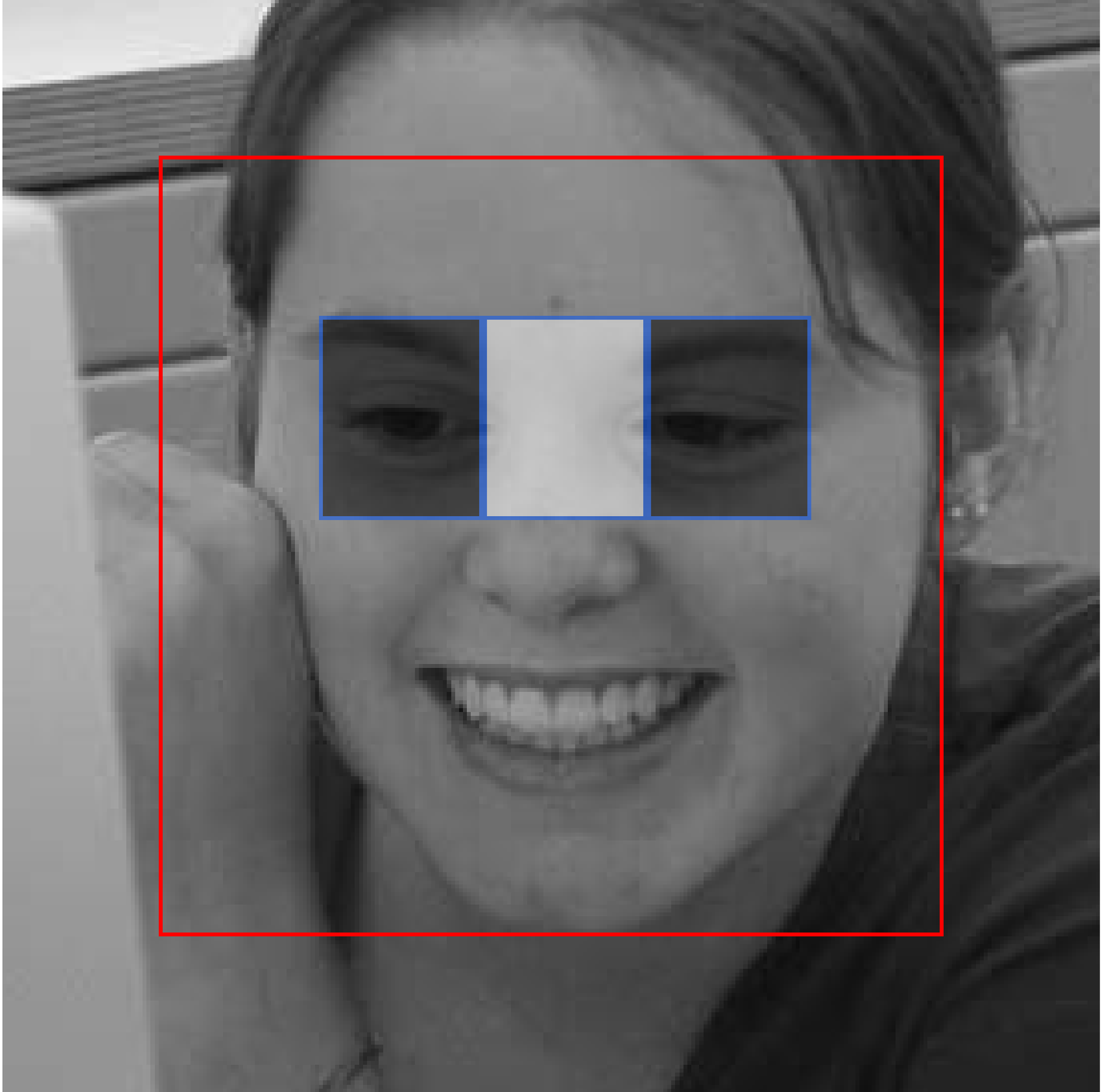

pain makes
pain makes
pain makes
pain makes
pain makes

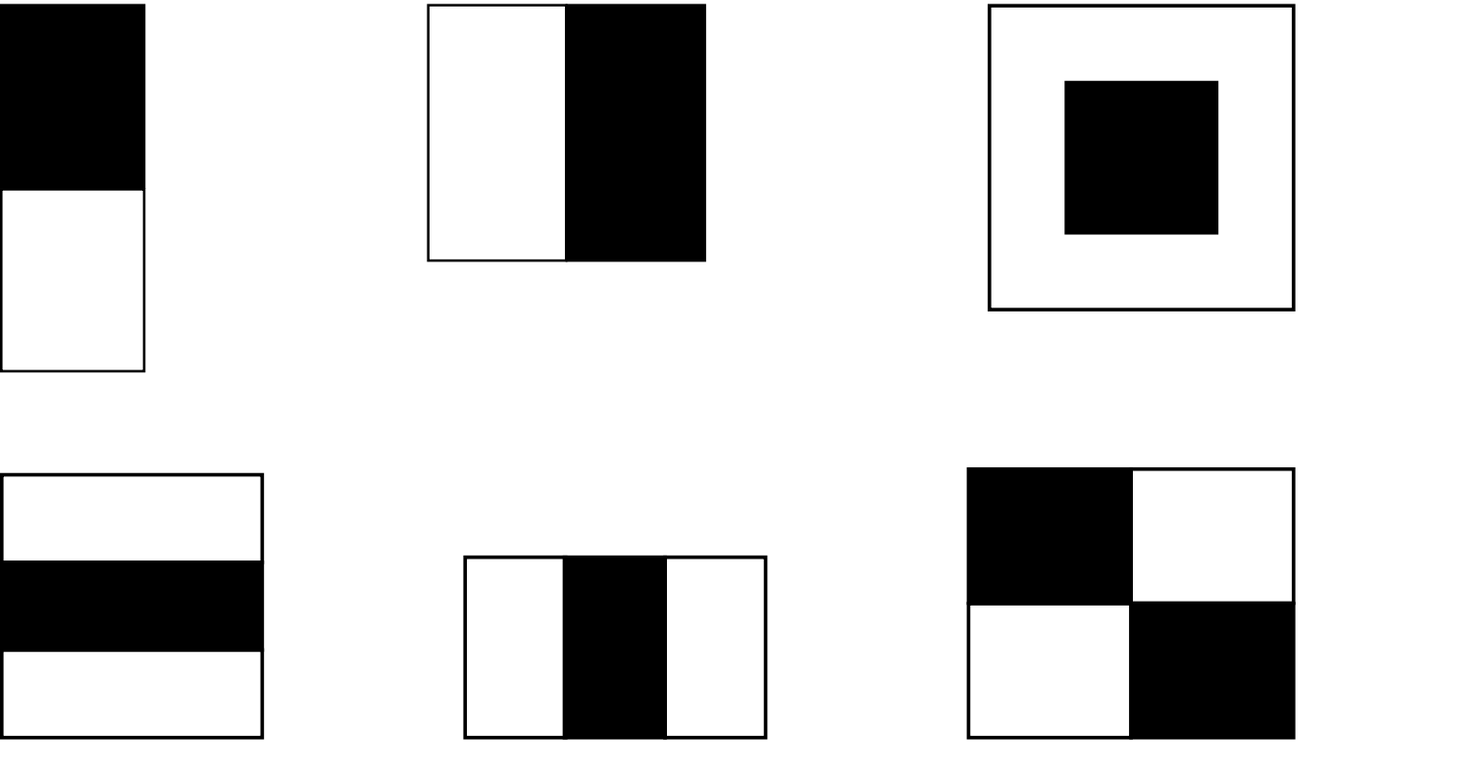
FACE

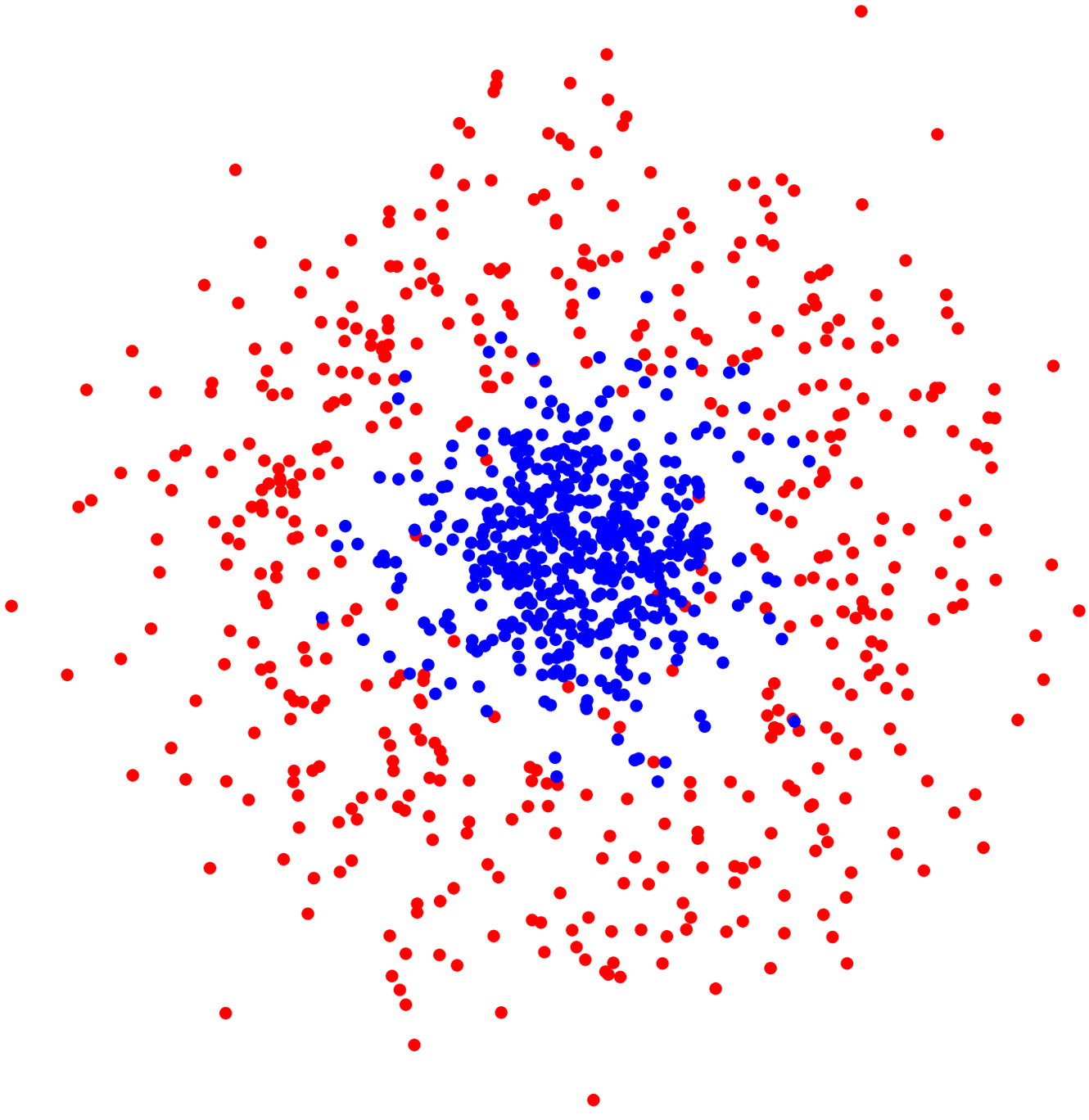
NONFACE

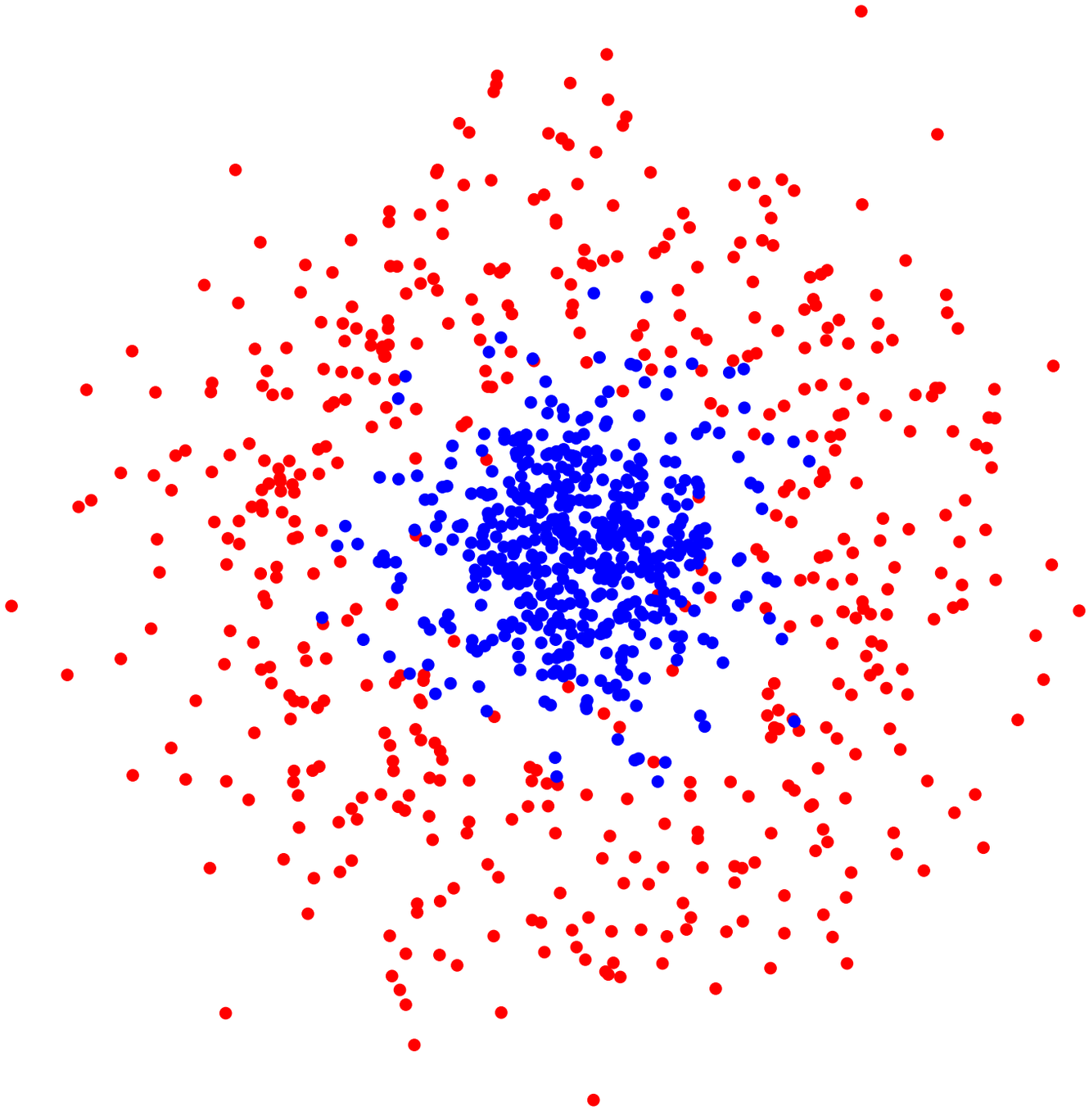

−1
−1
−1
−1
$S_1(x)$
#
$S_2(x)$
#
$S_3(x)$
#
$S_n(x)$
+1
+1
+1
+1

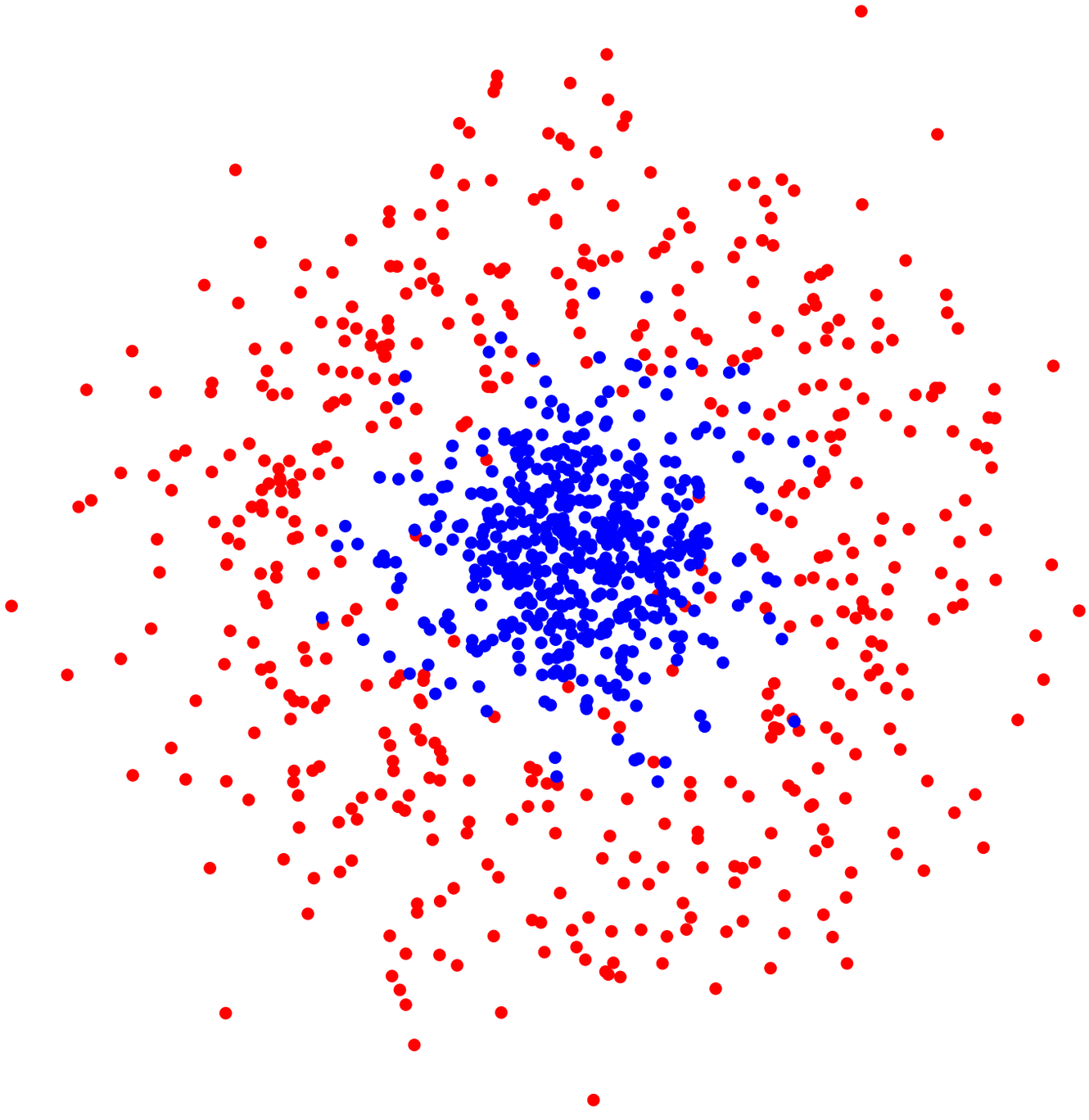


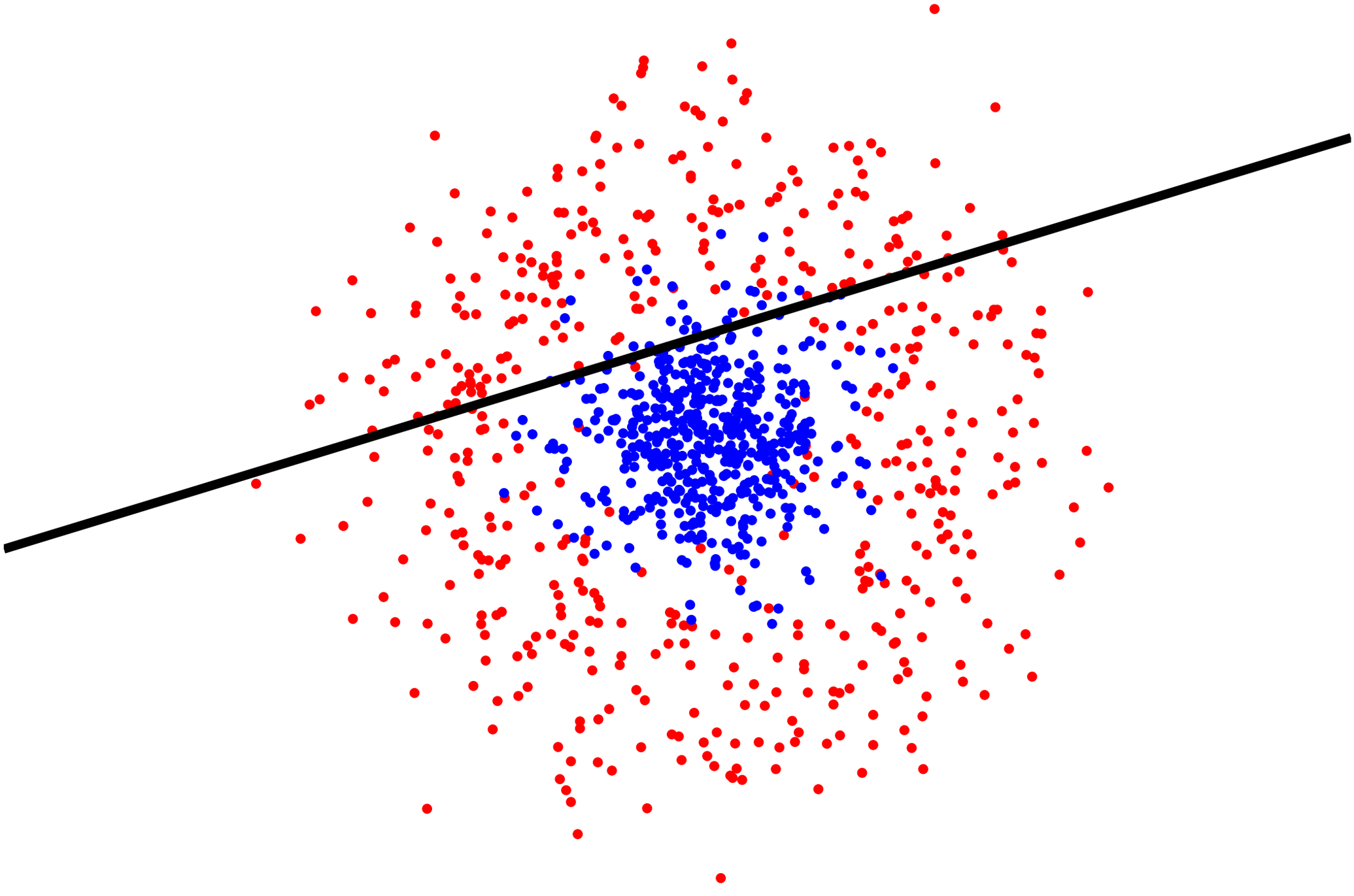


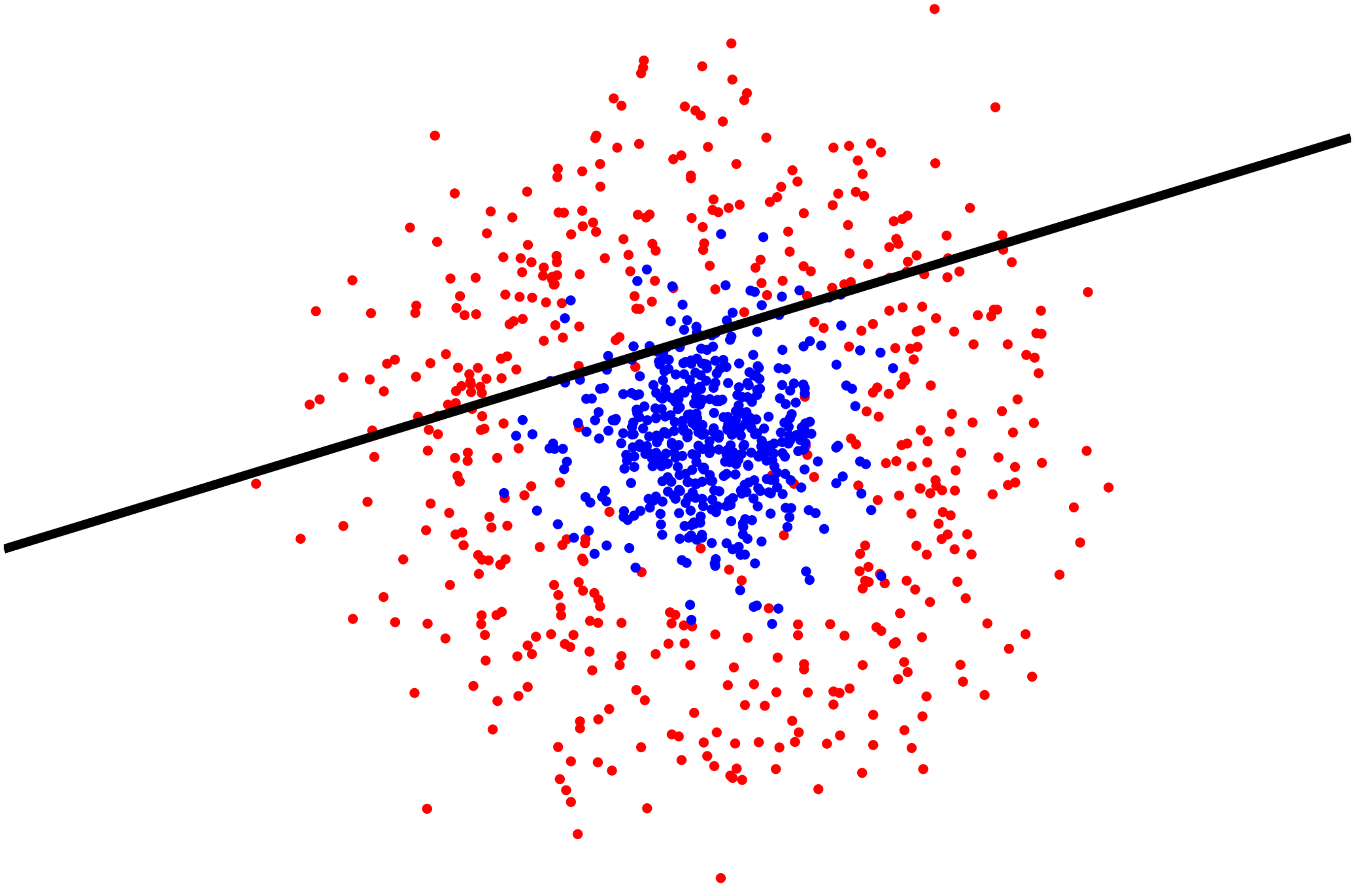

..... -1
..... +1

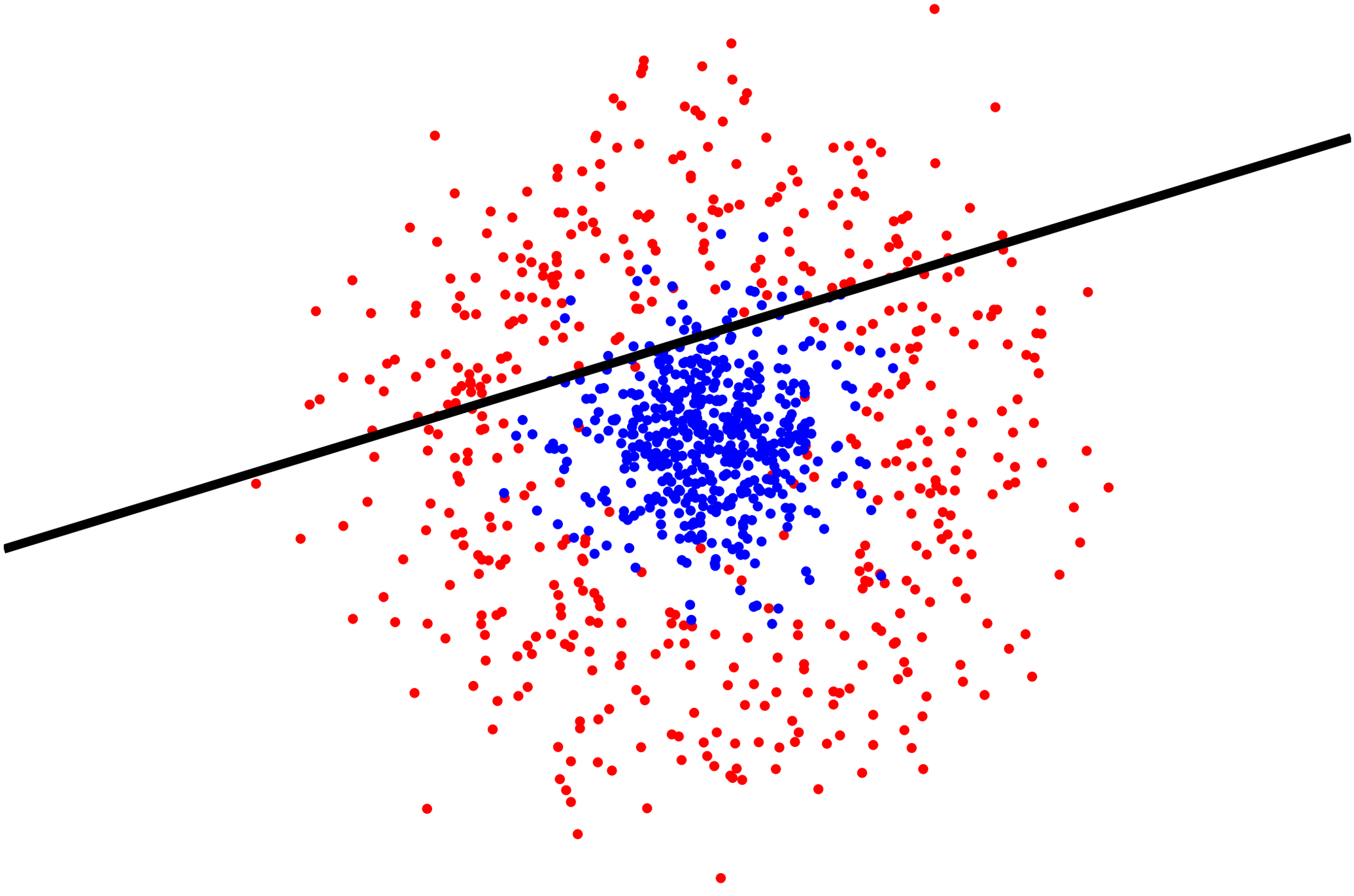


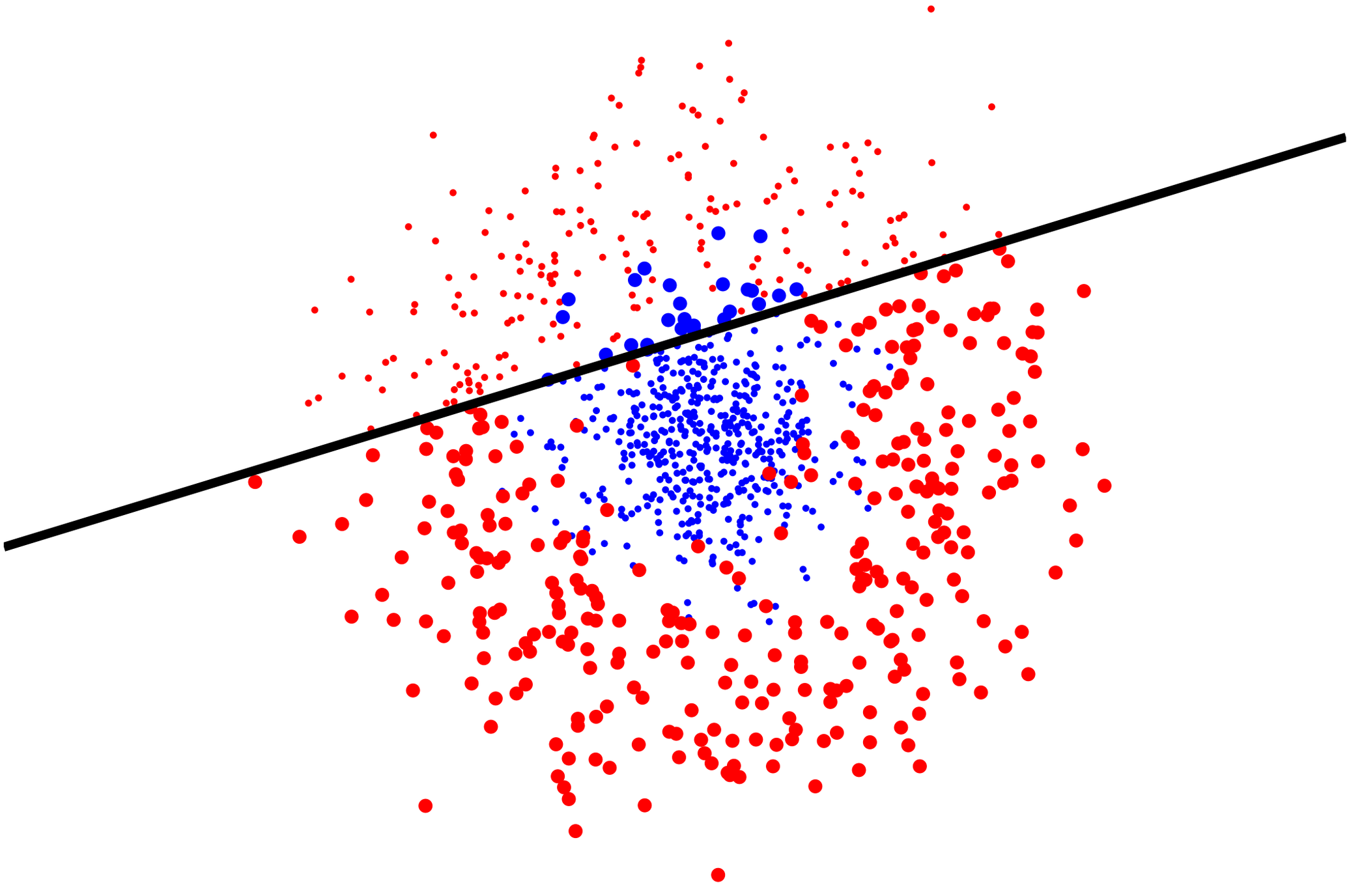


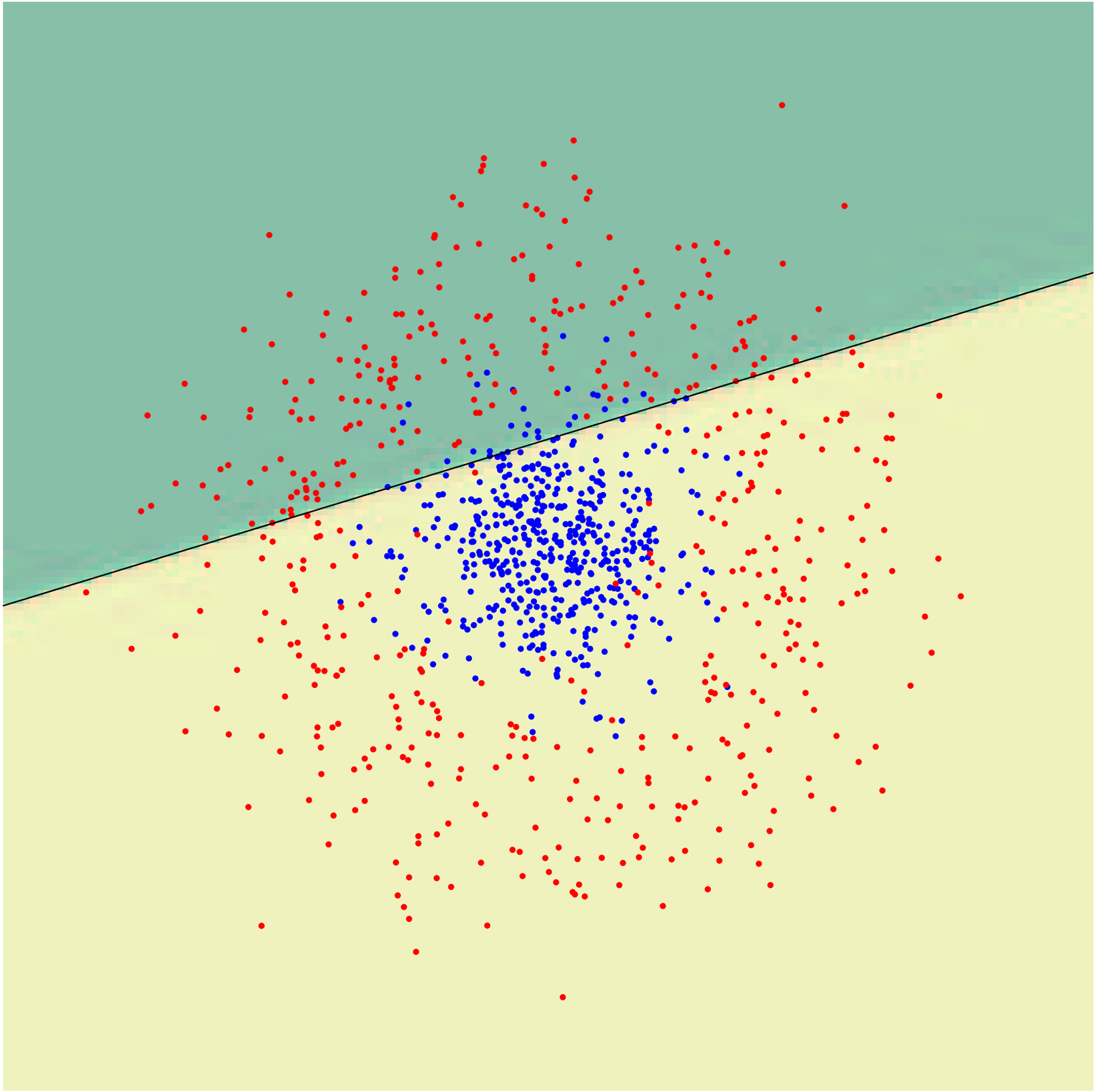


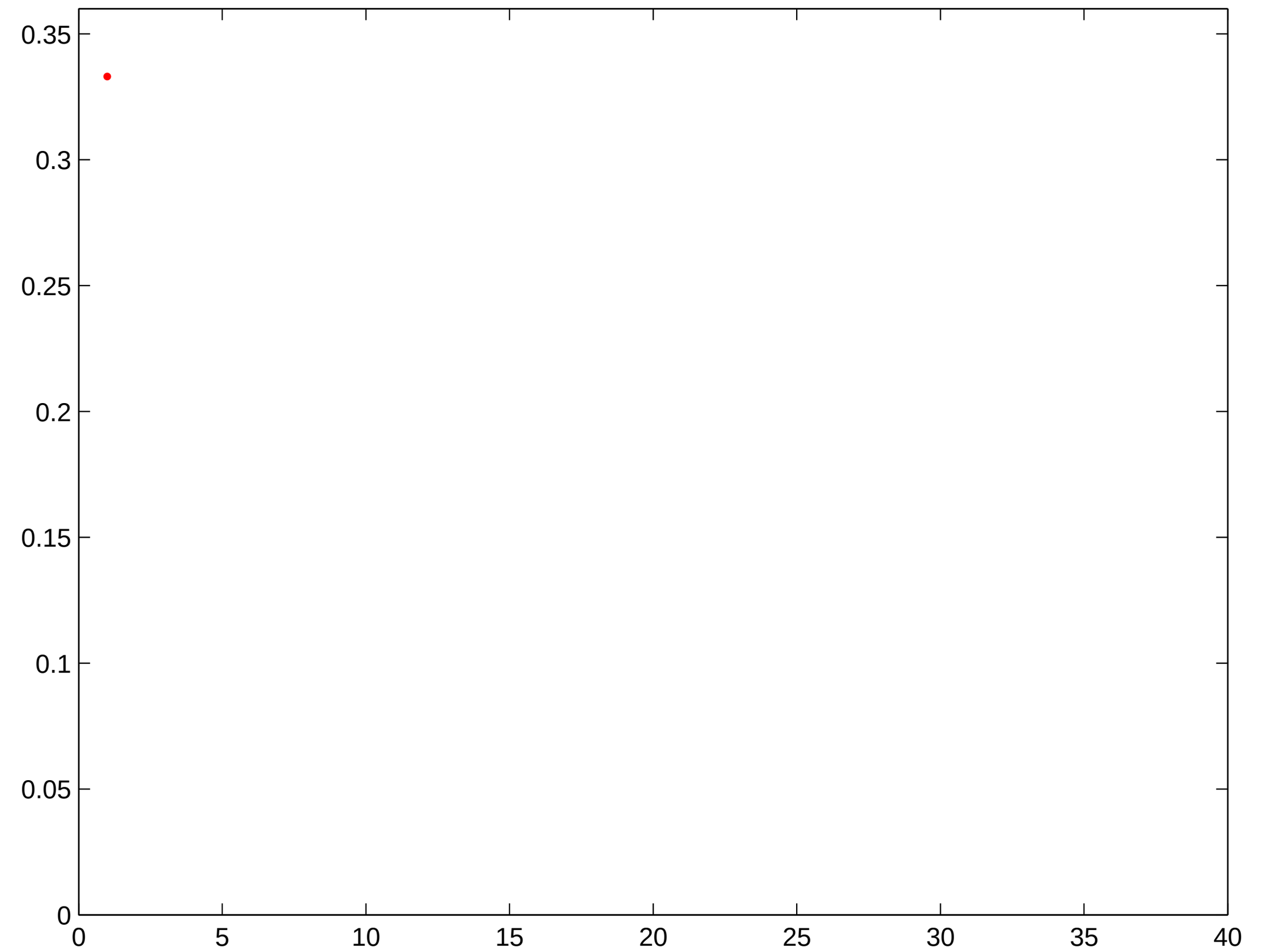


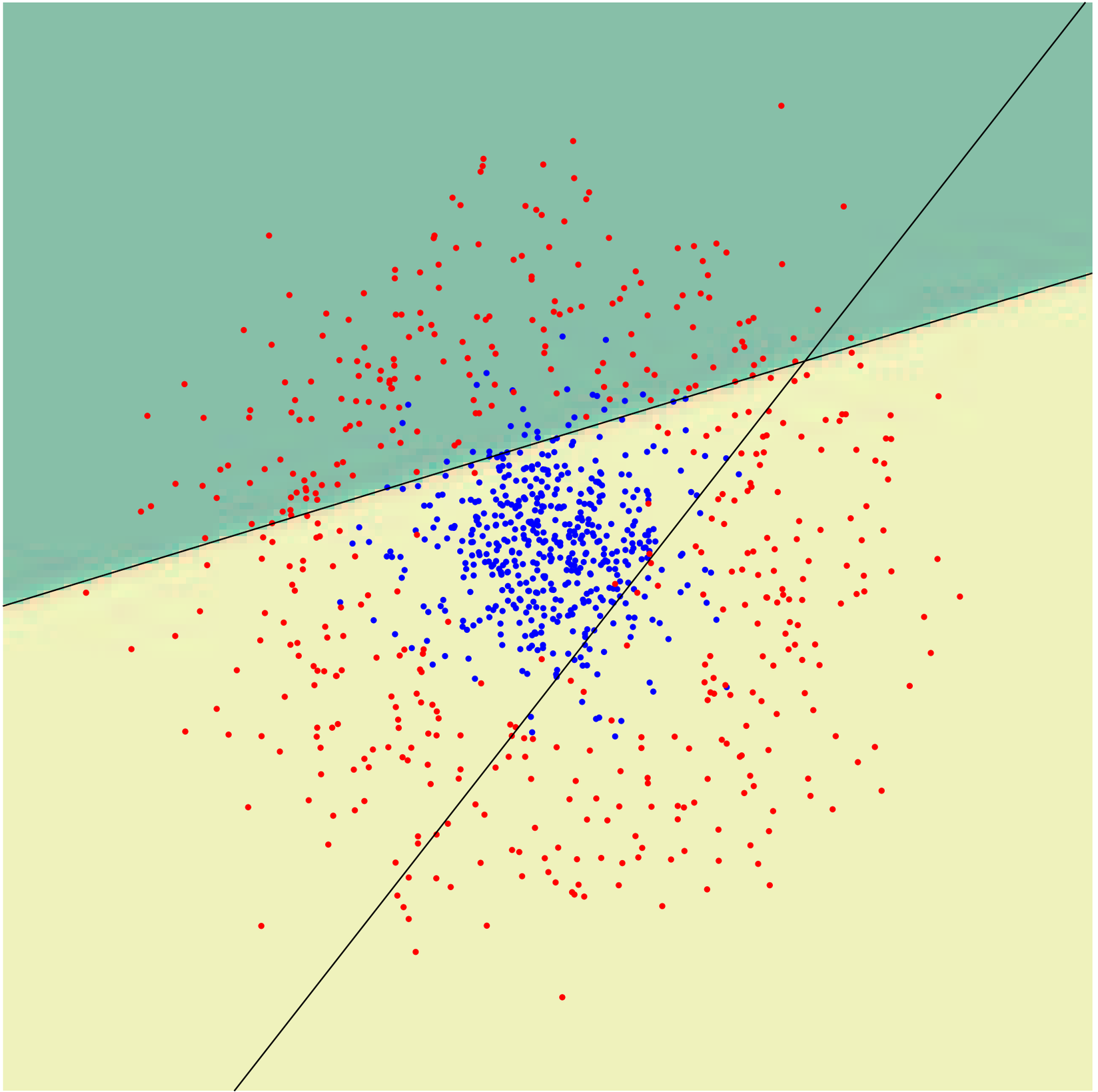


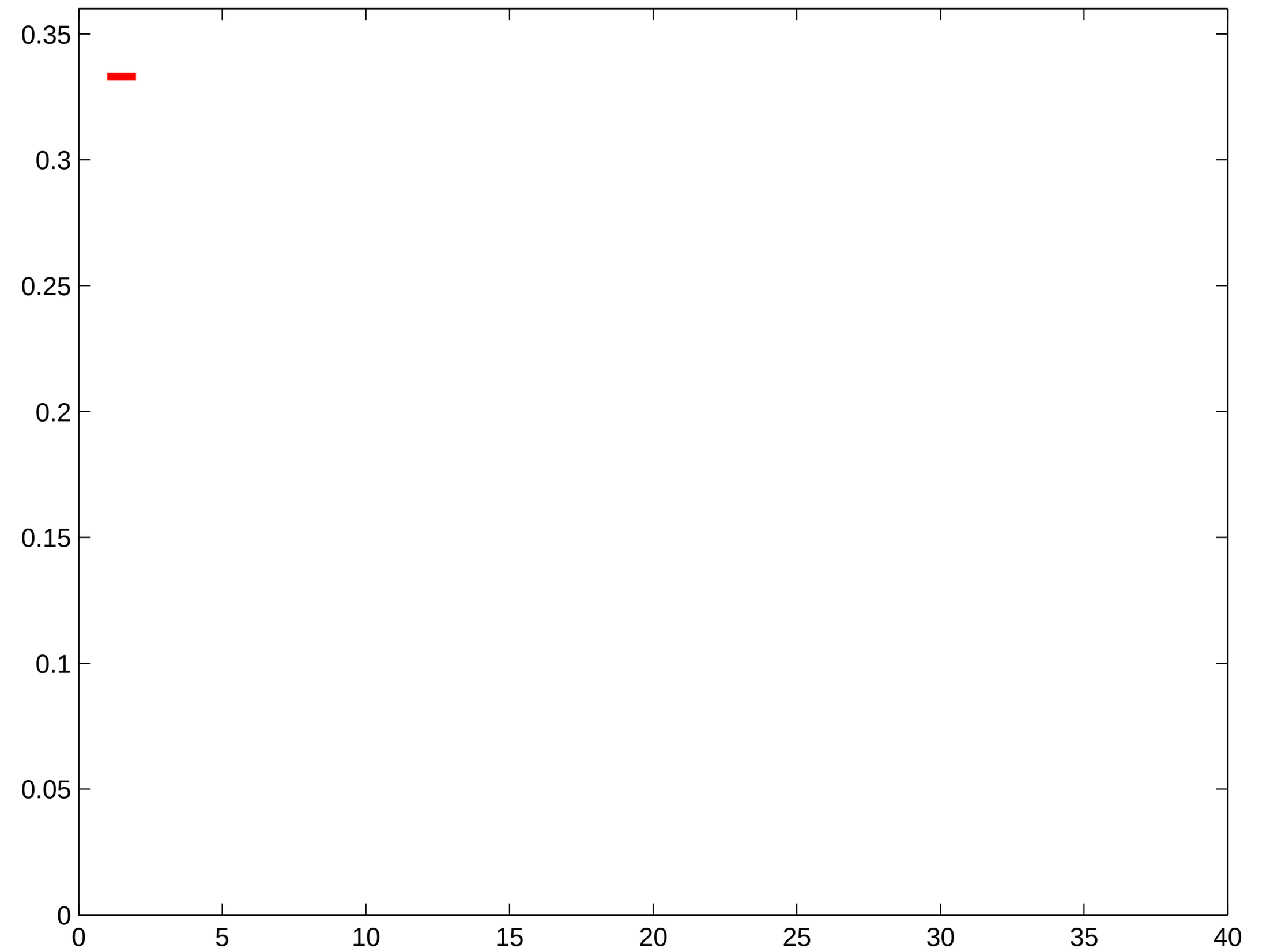


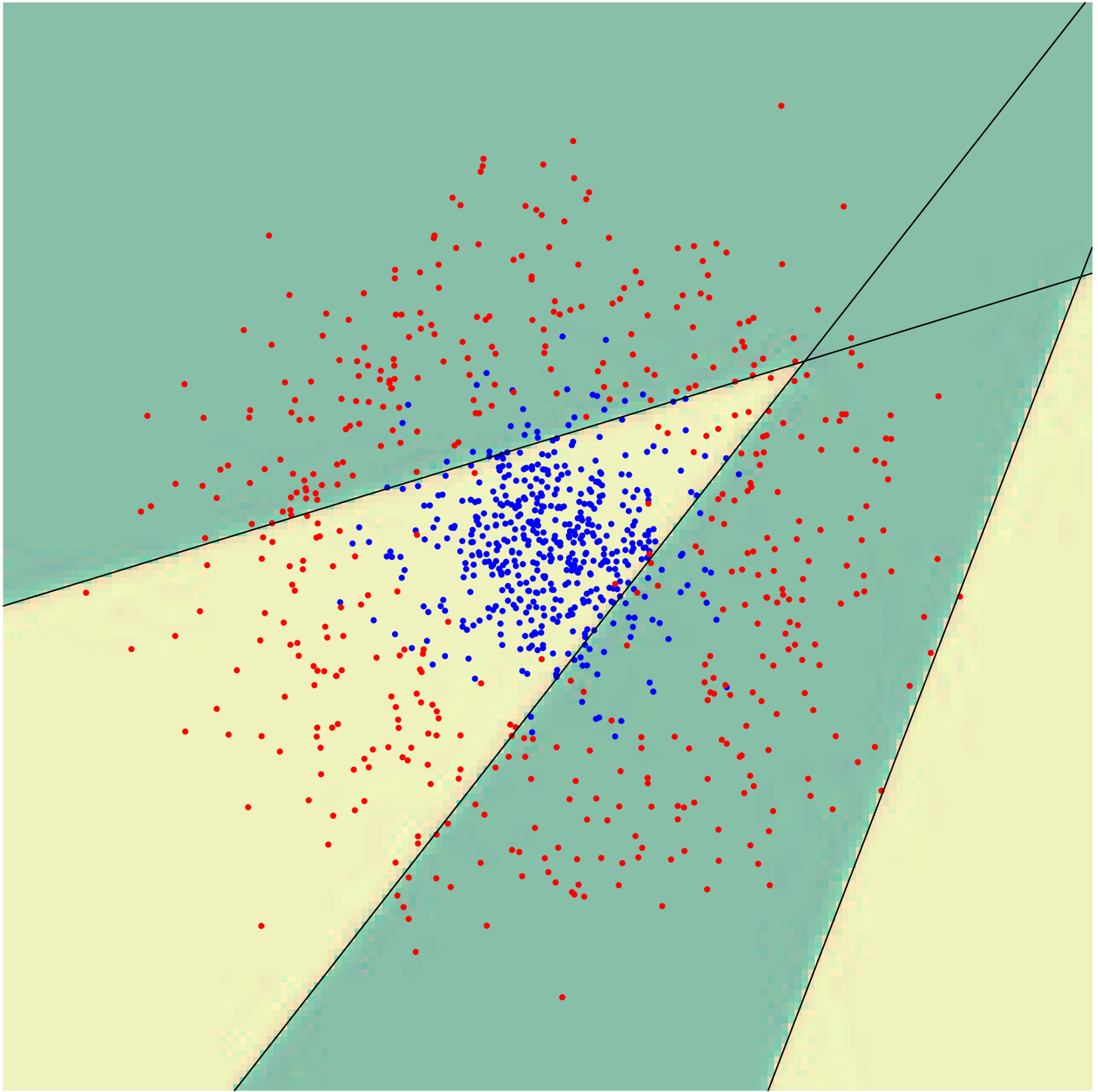


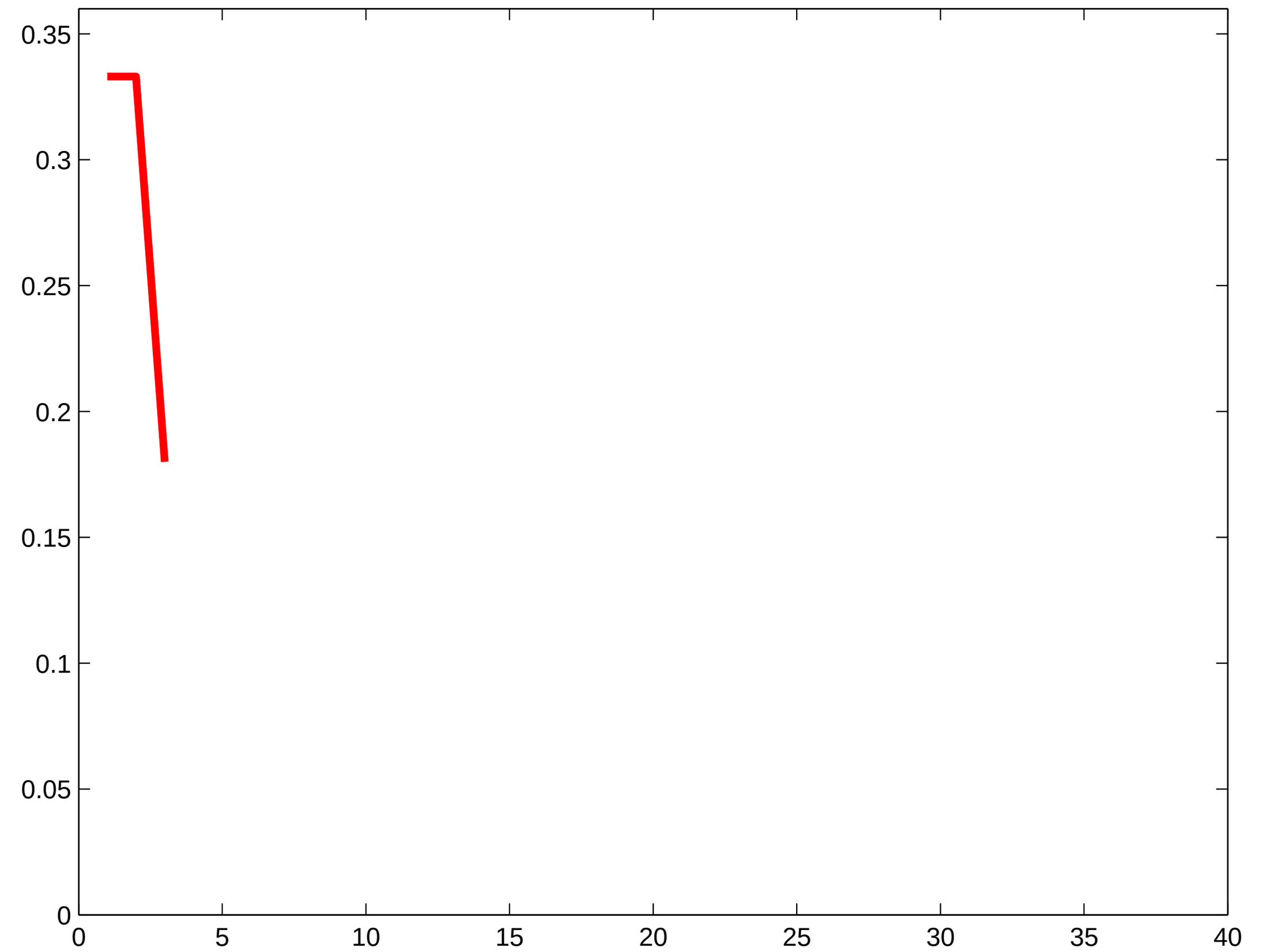


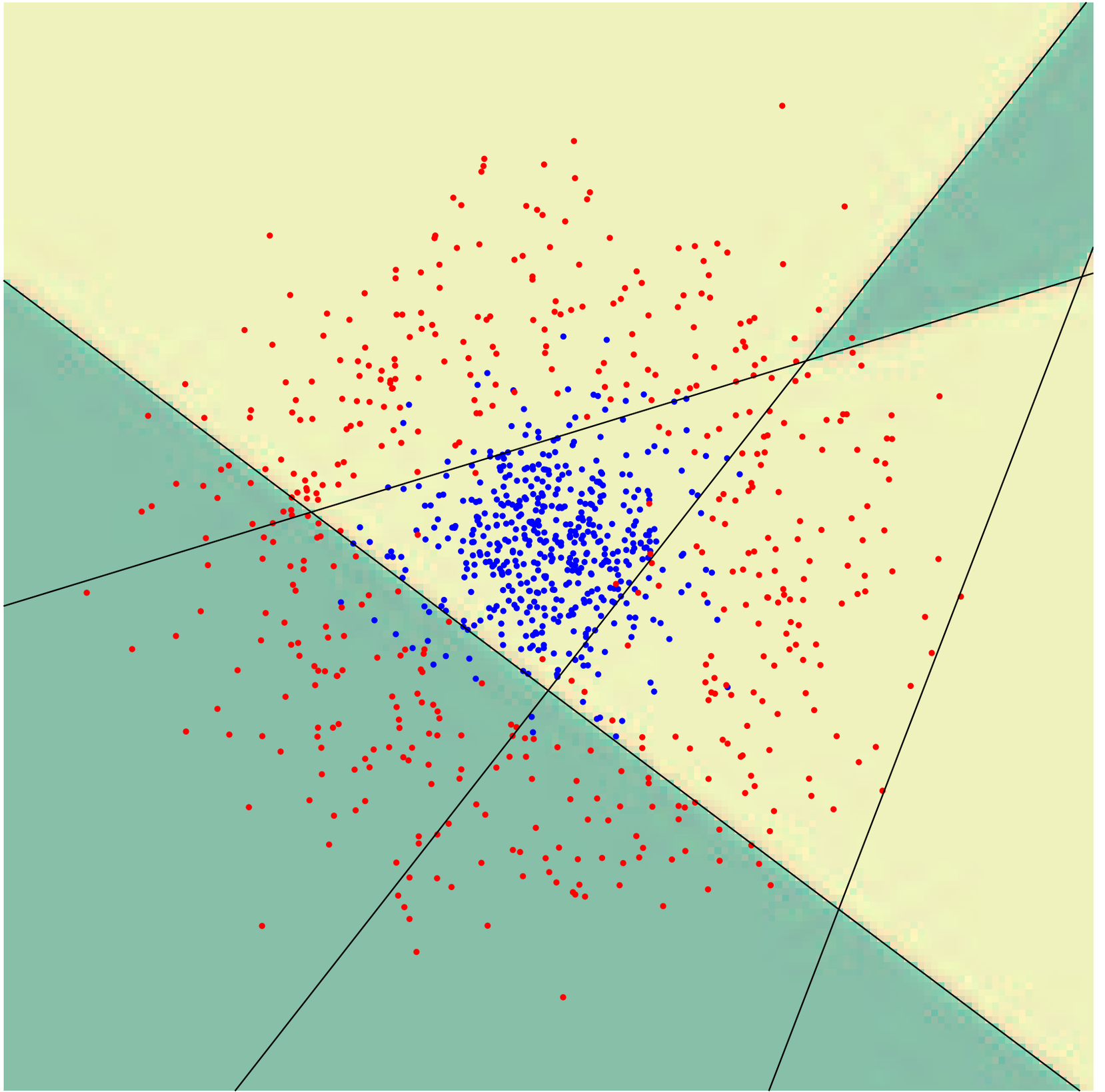

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40

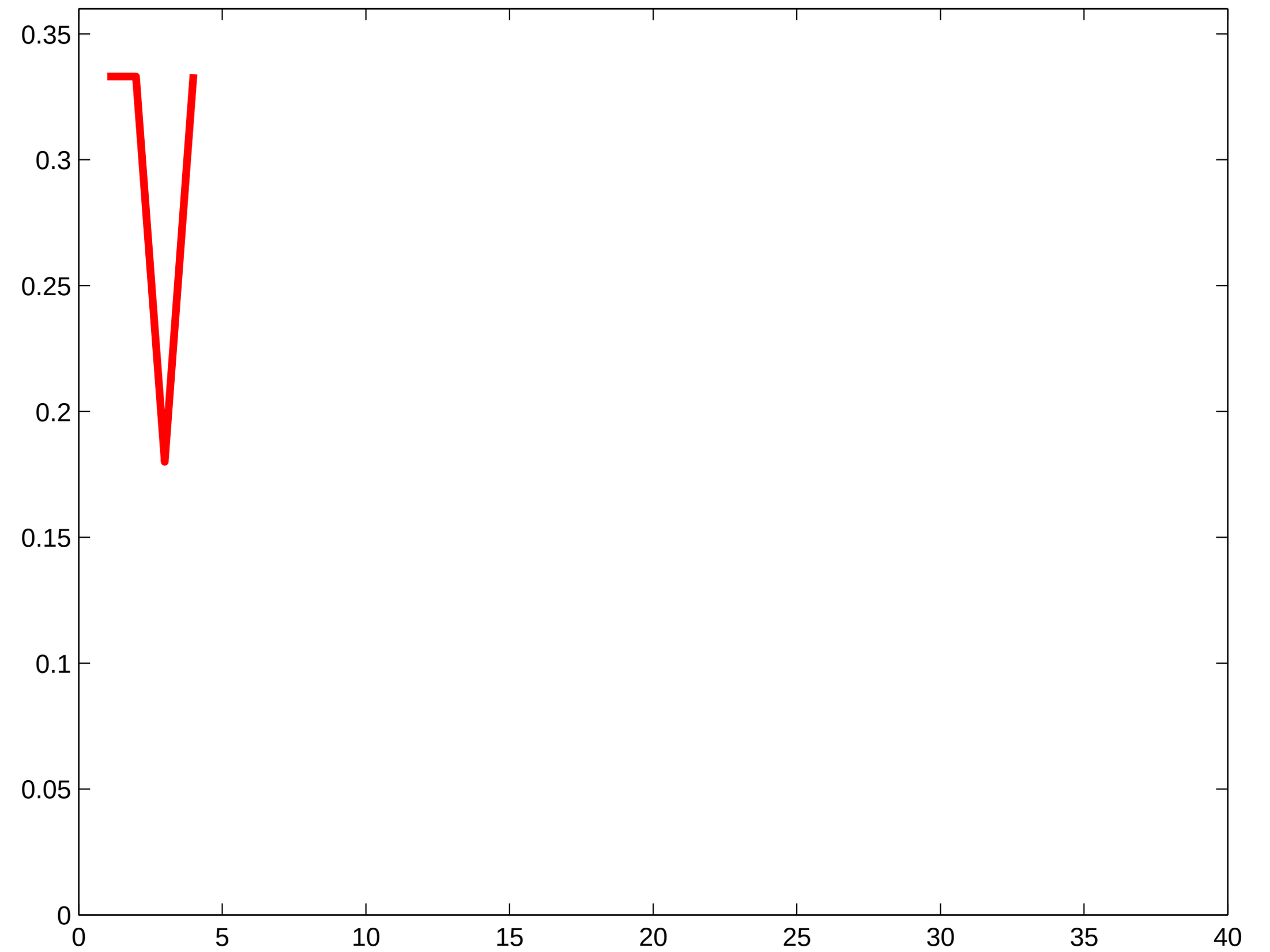


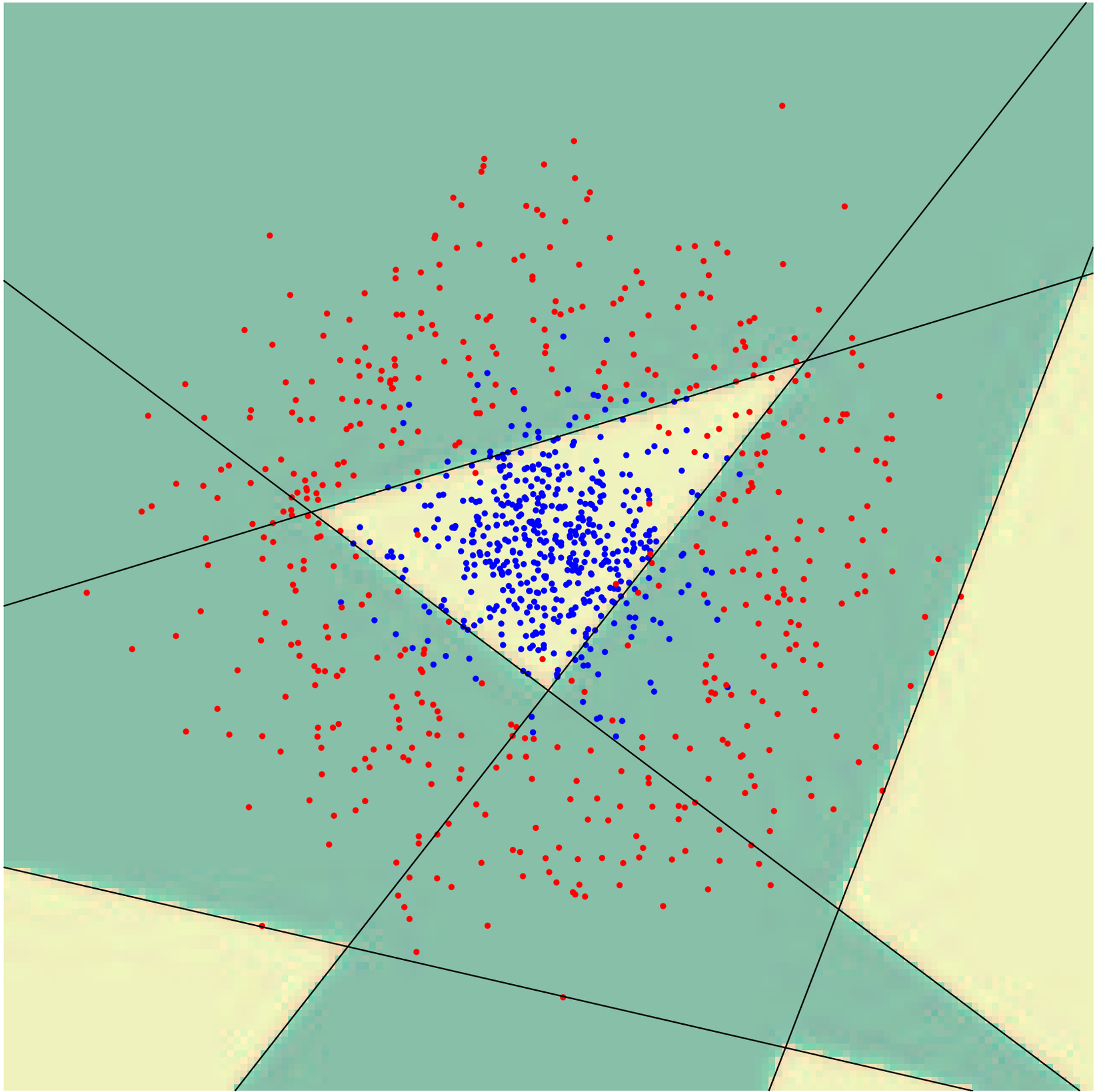

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40

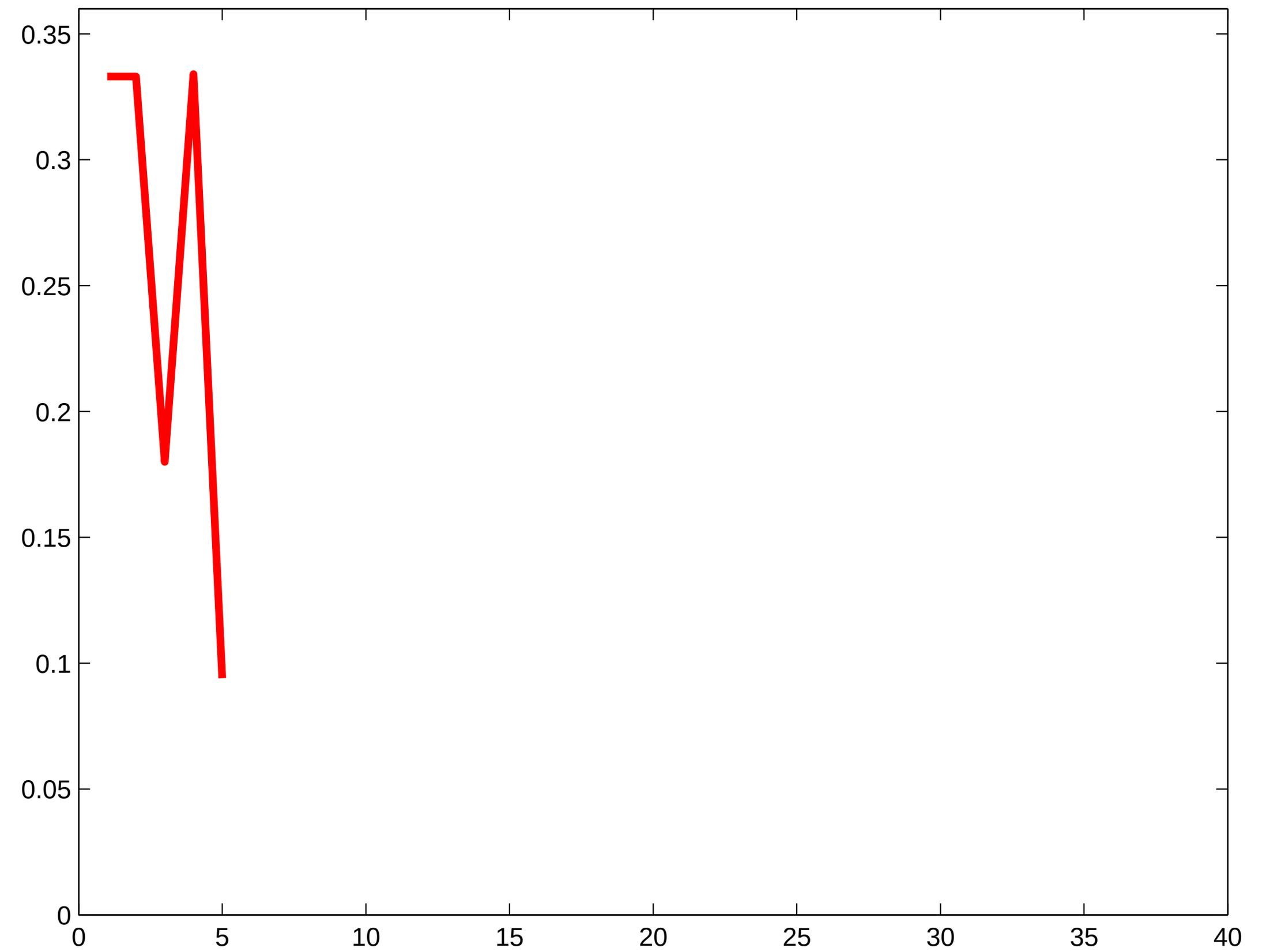


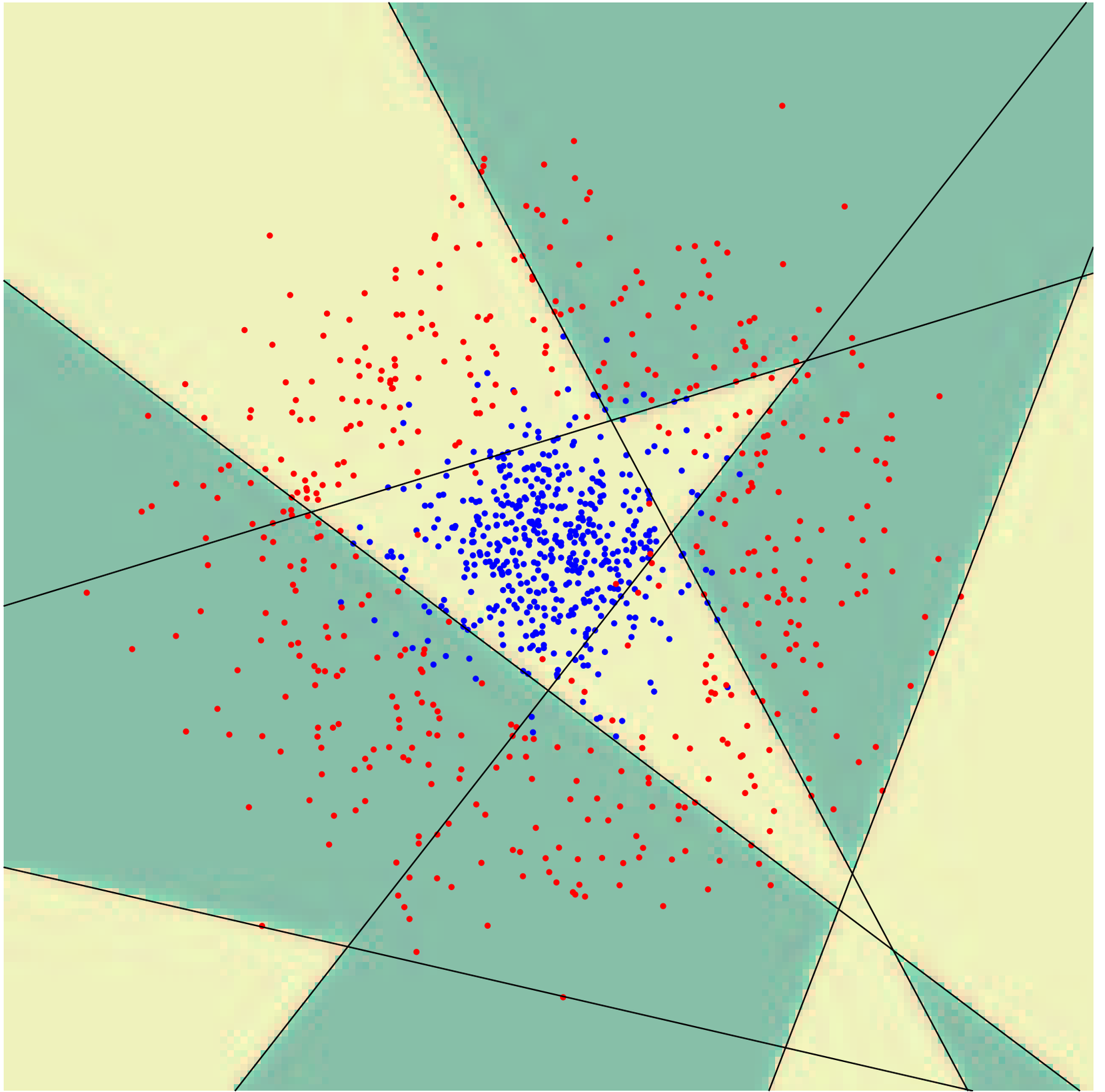

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40

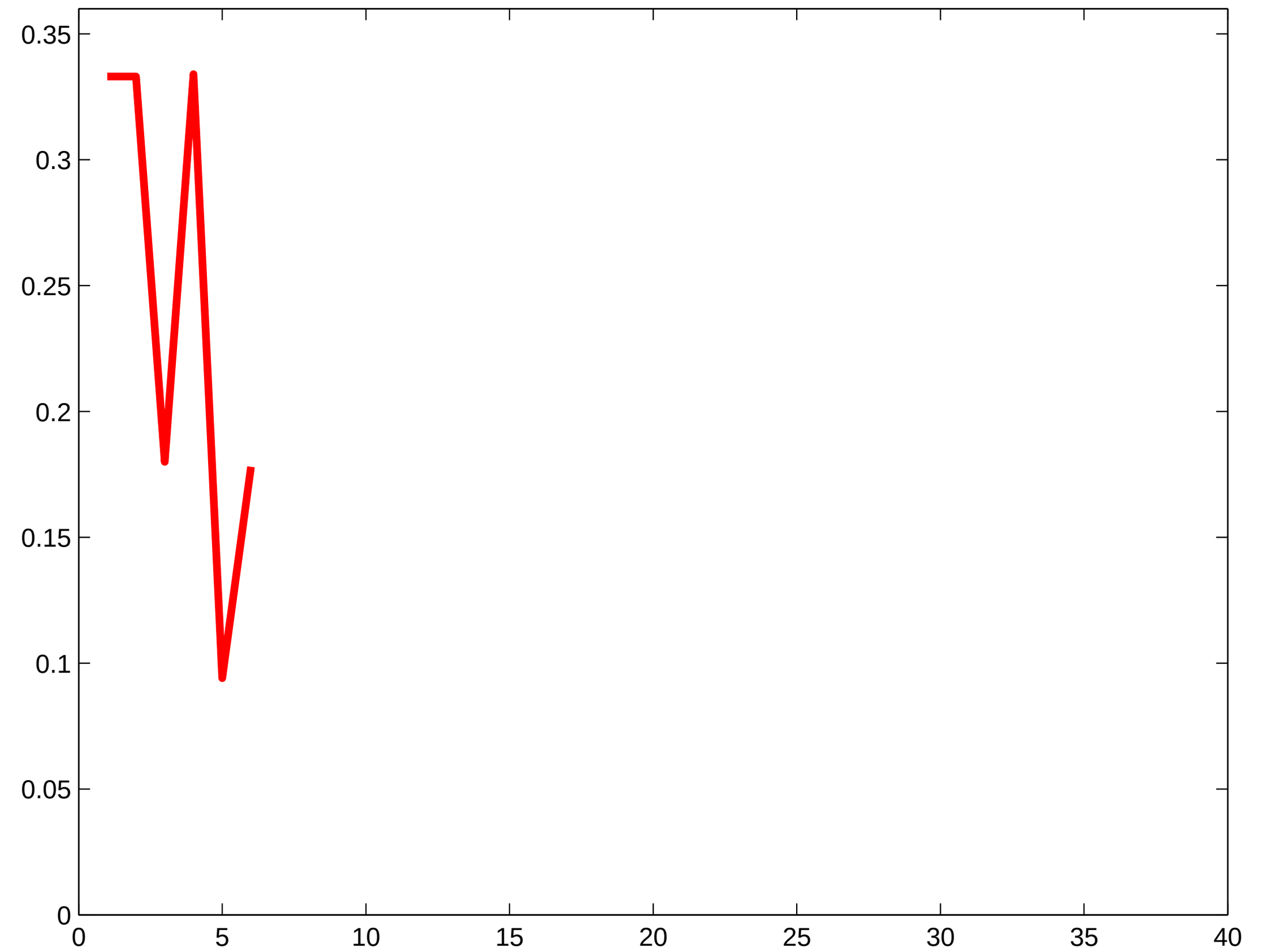


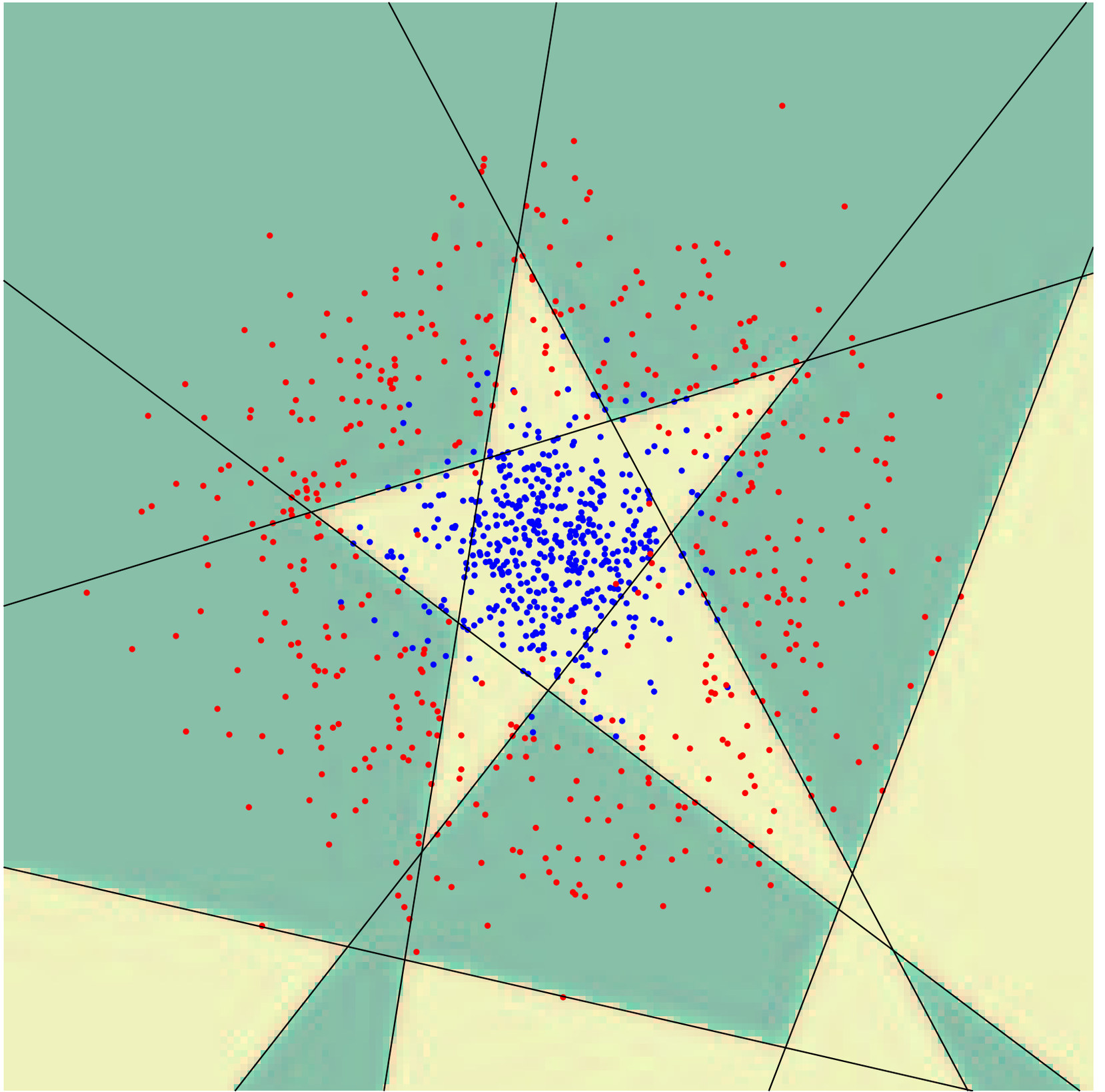

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40

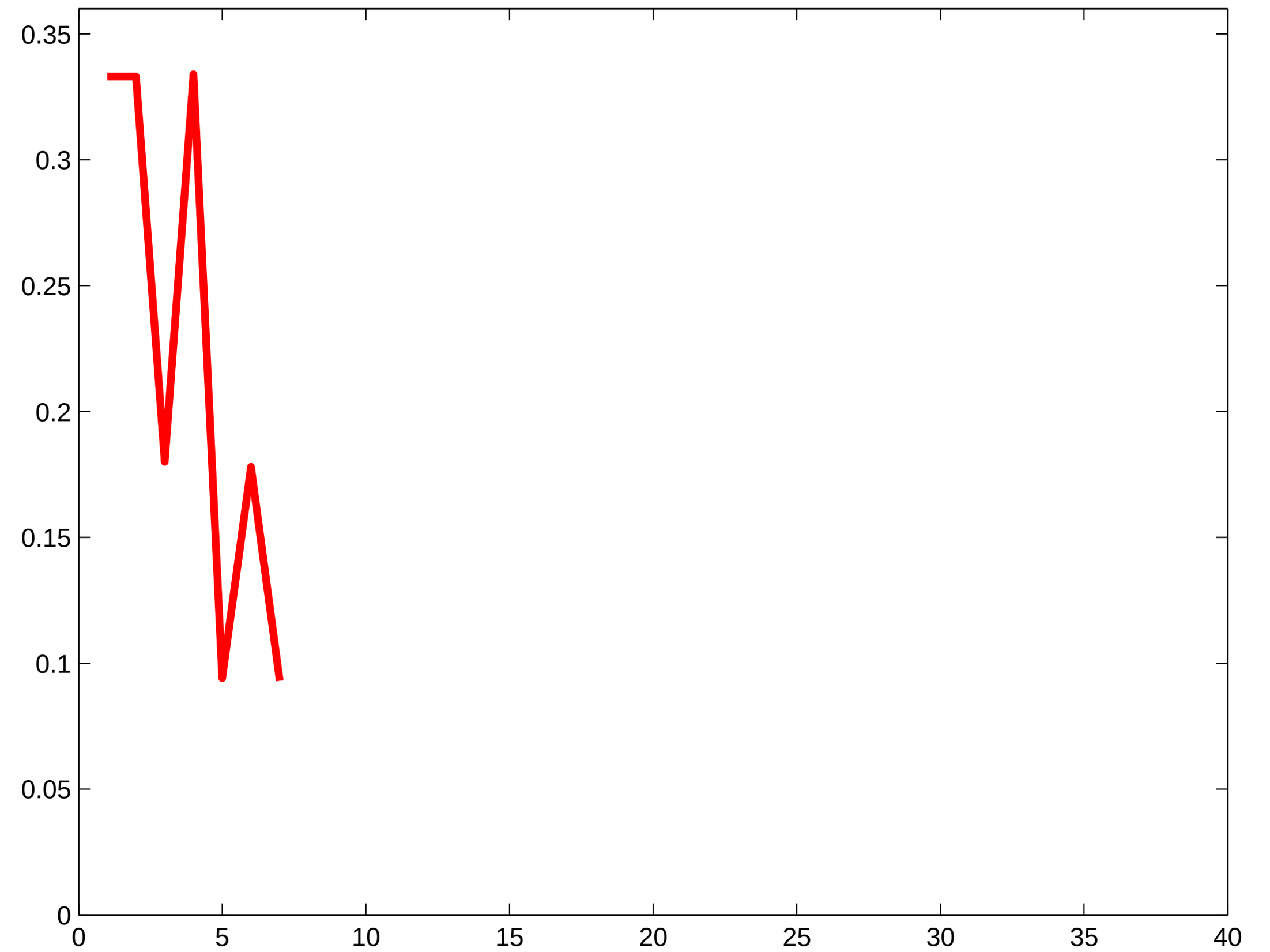


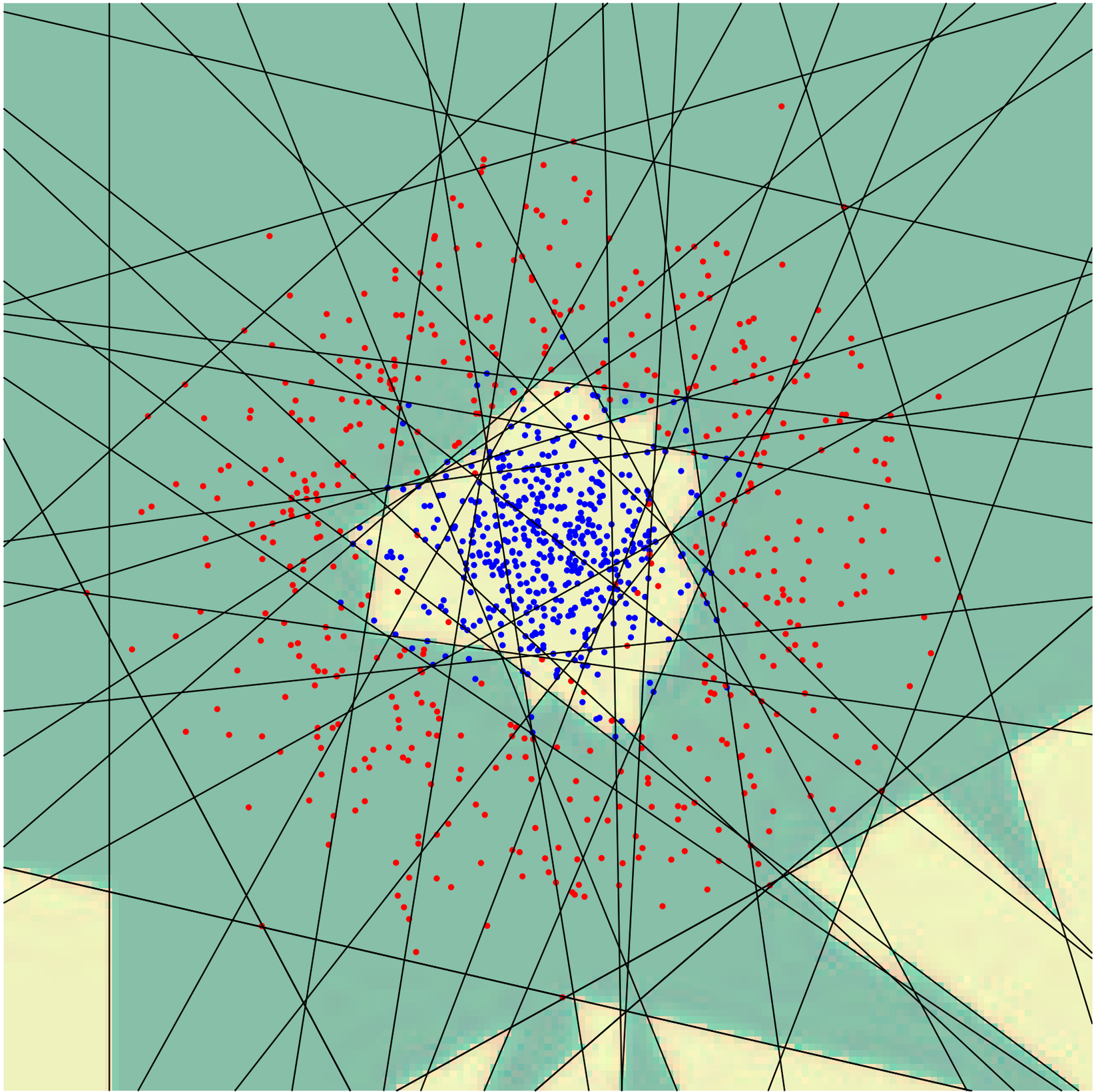

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40

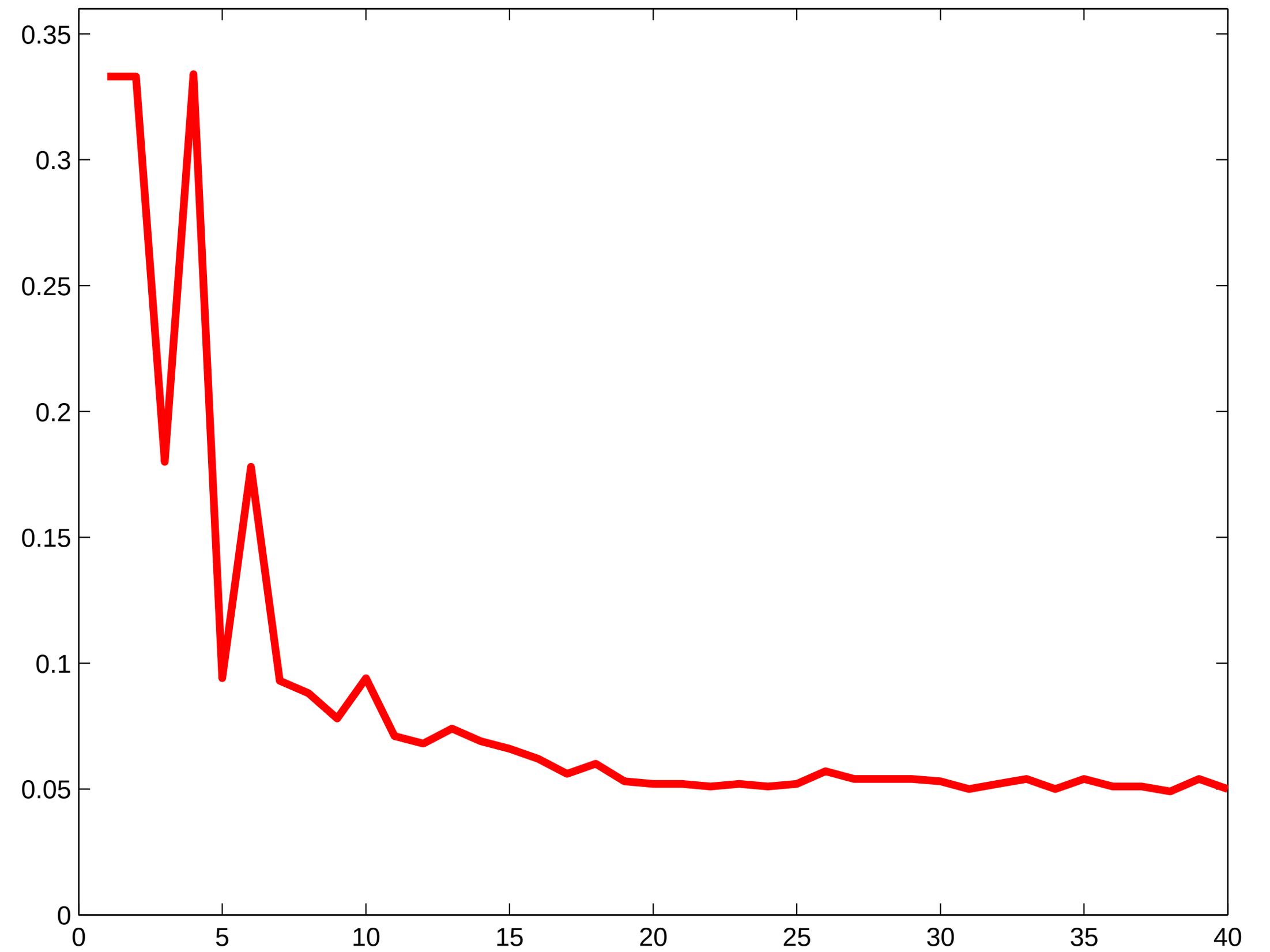


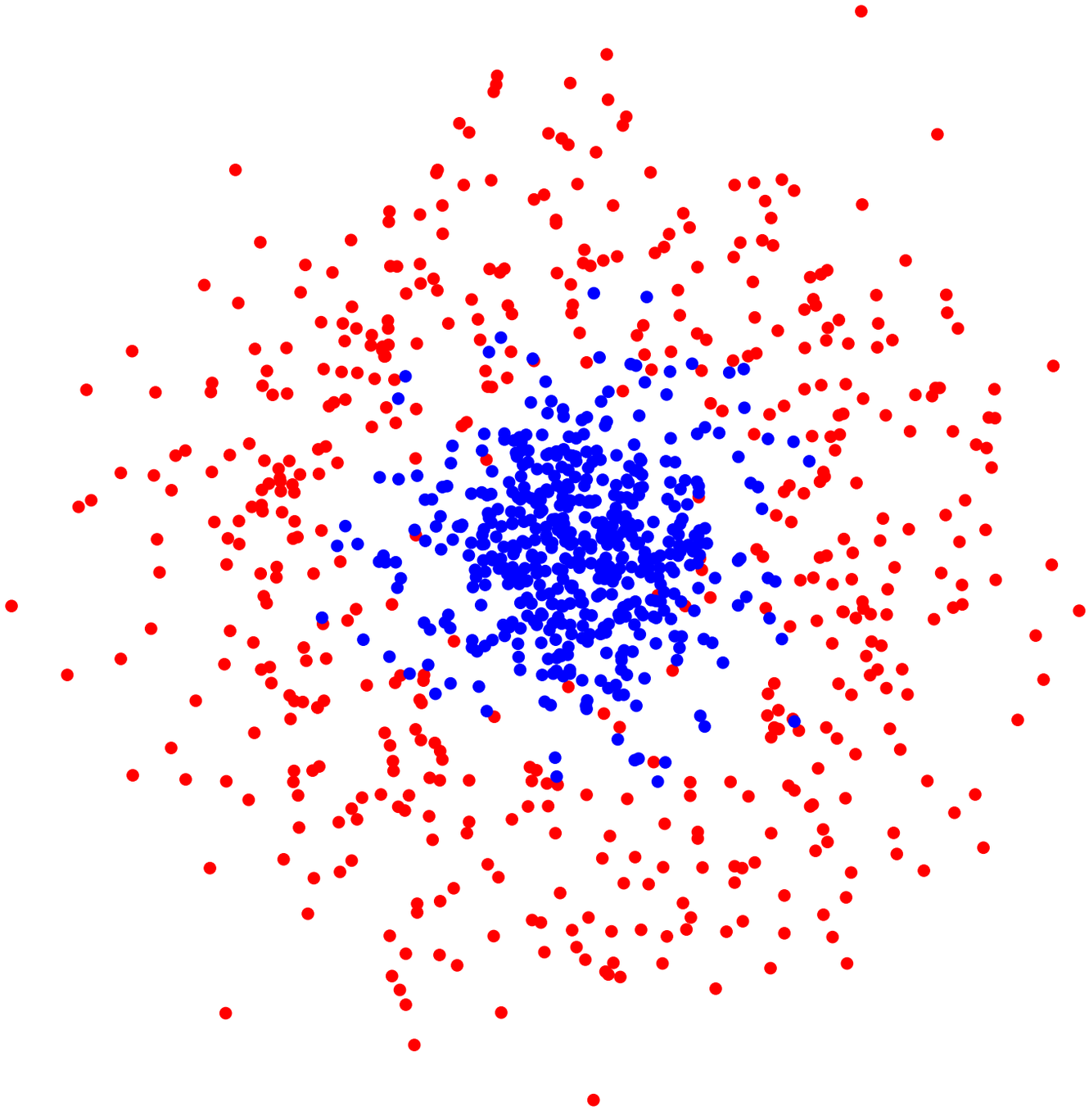

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40

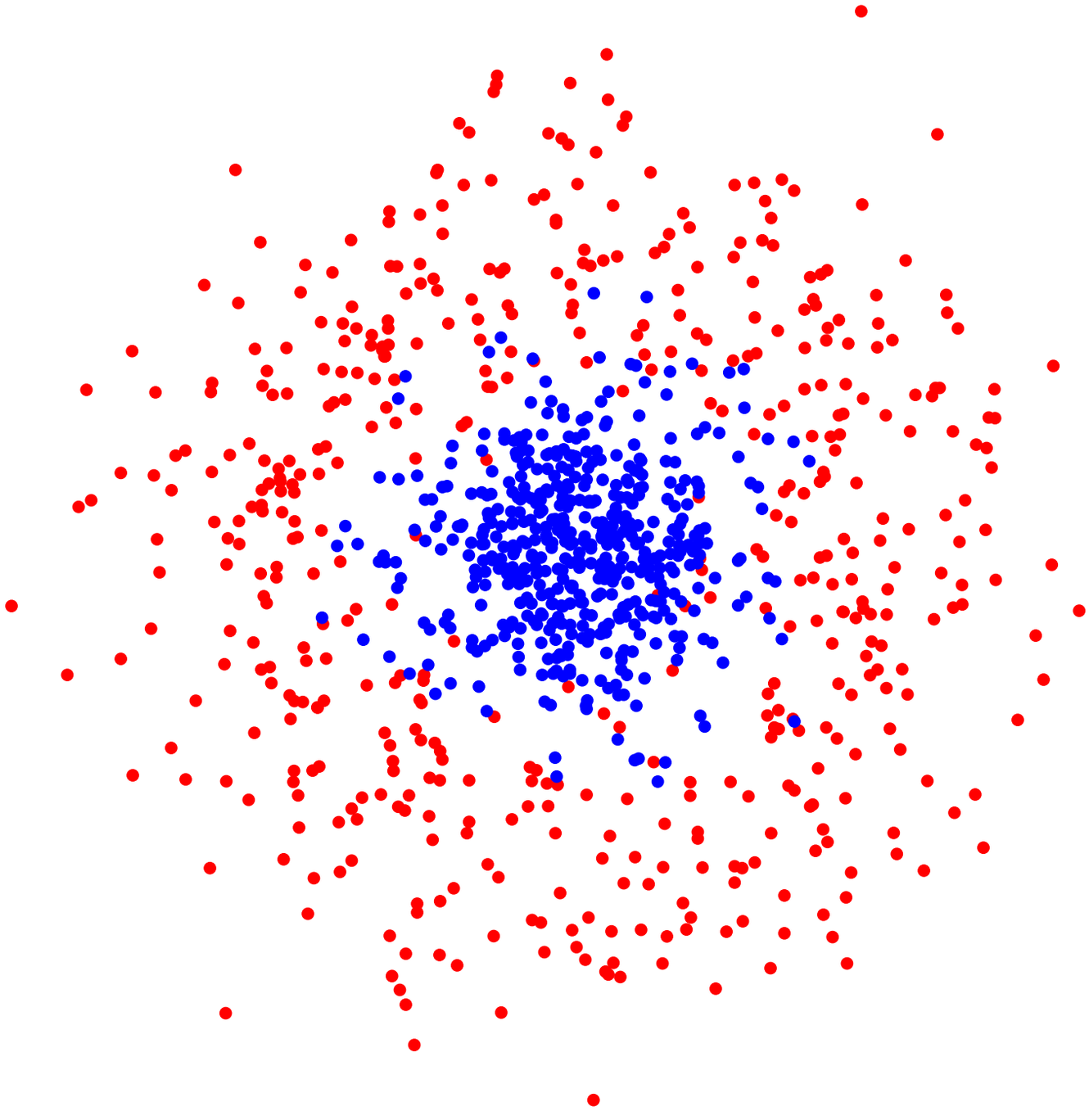


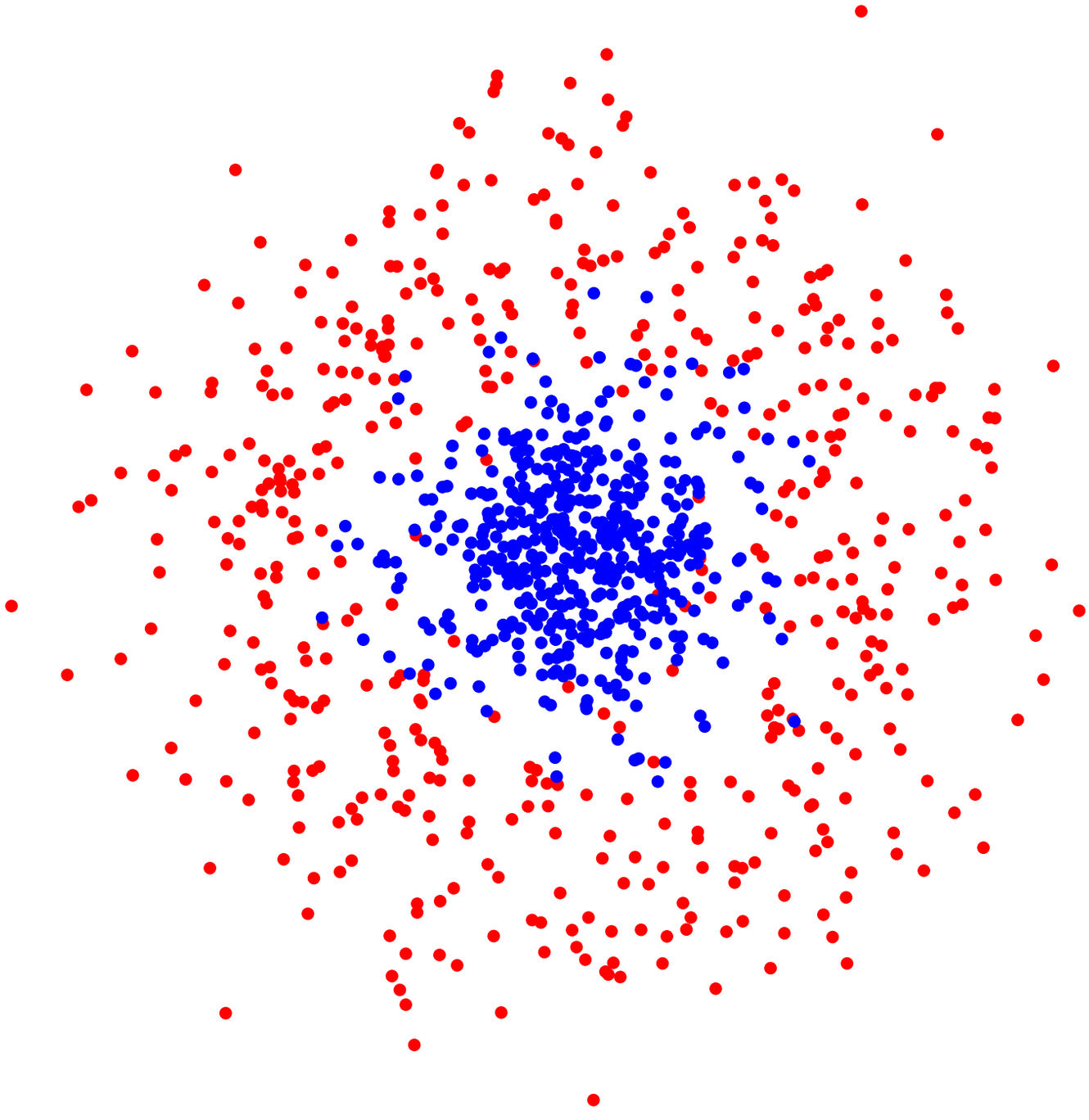

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40

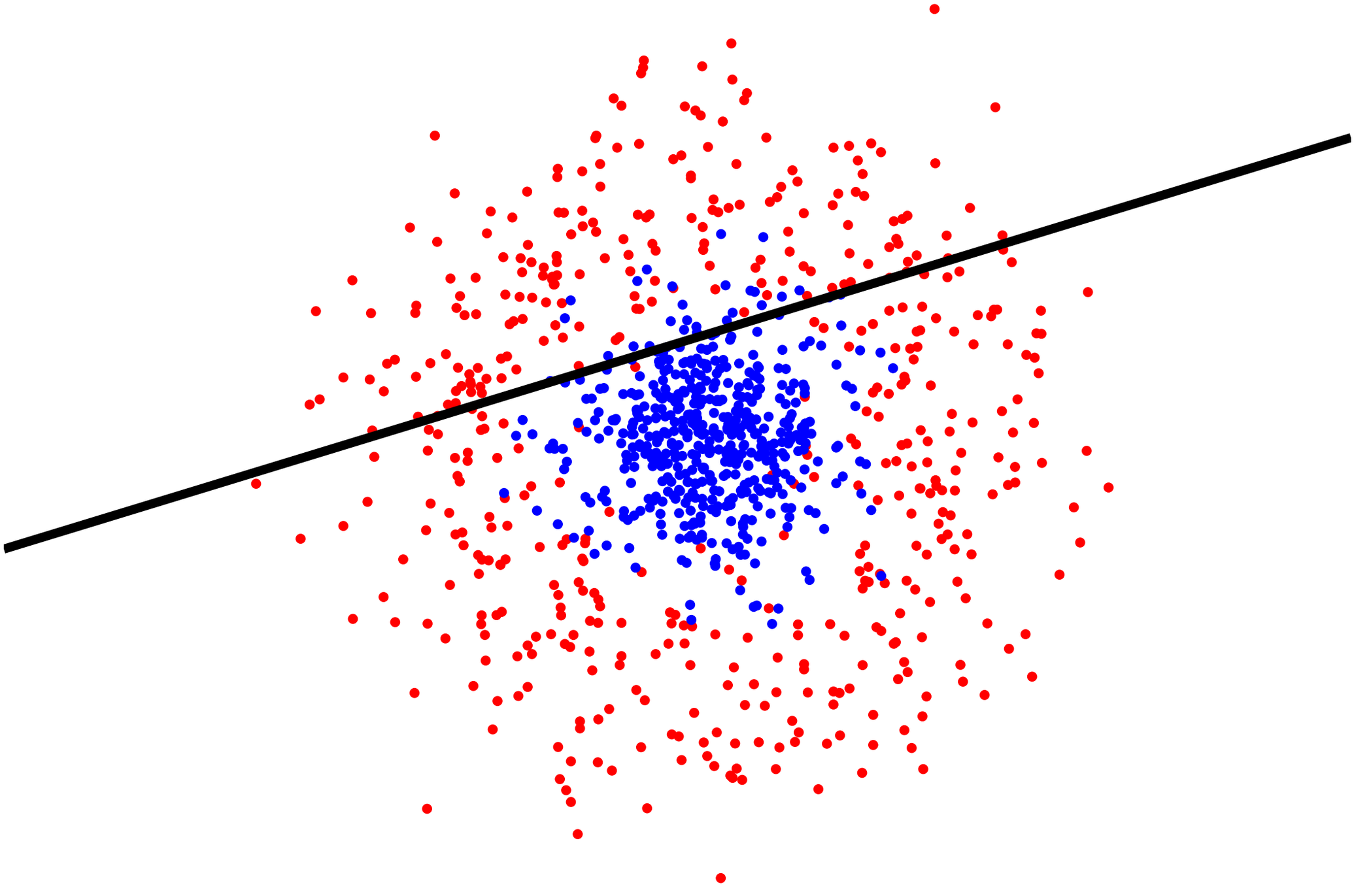


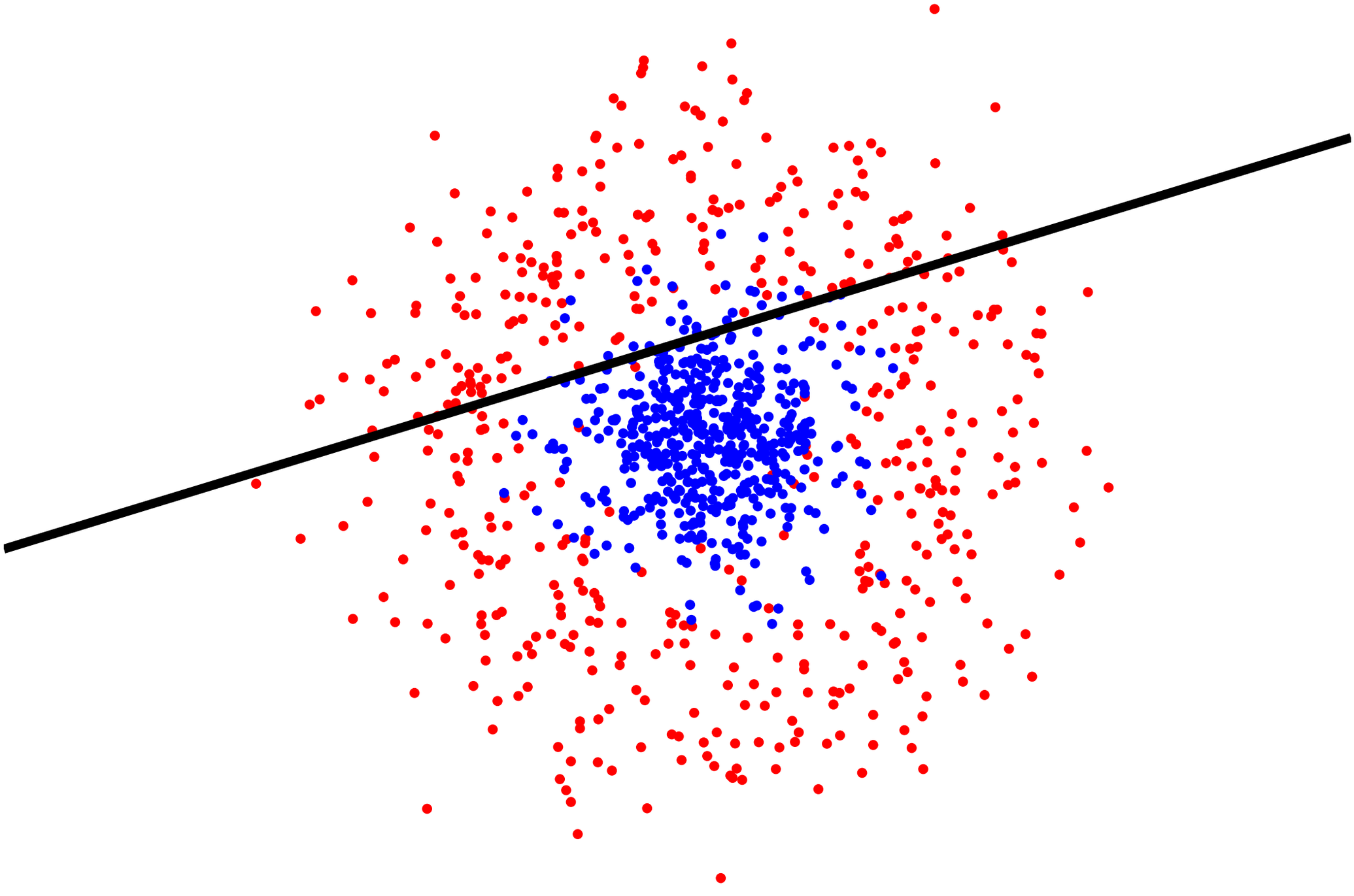

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40

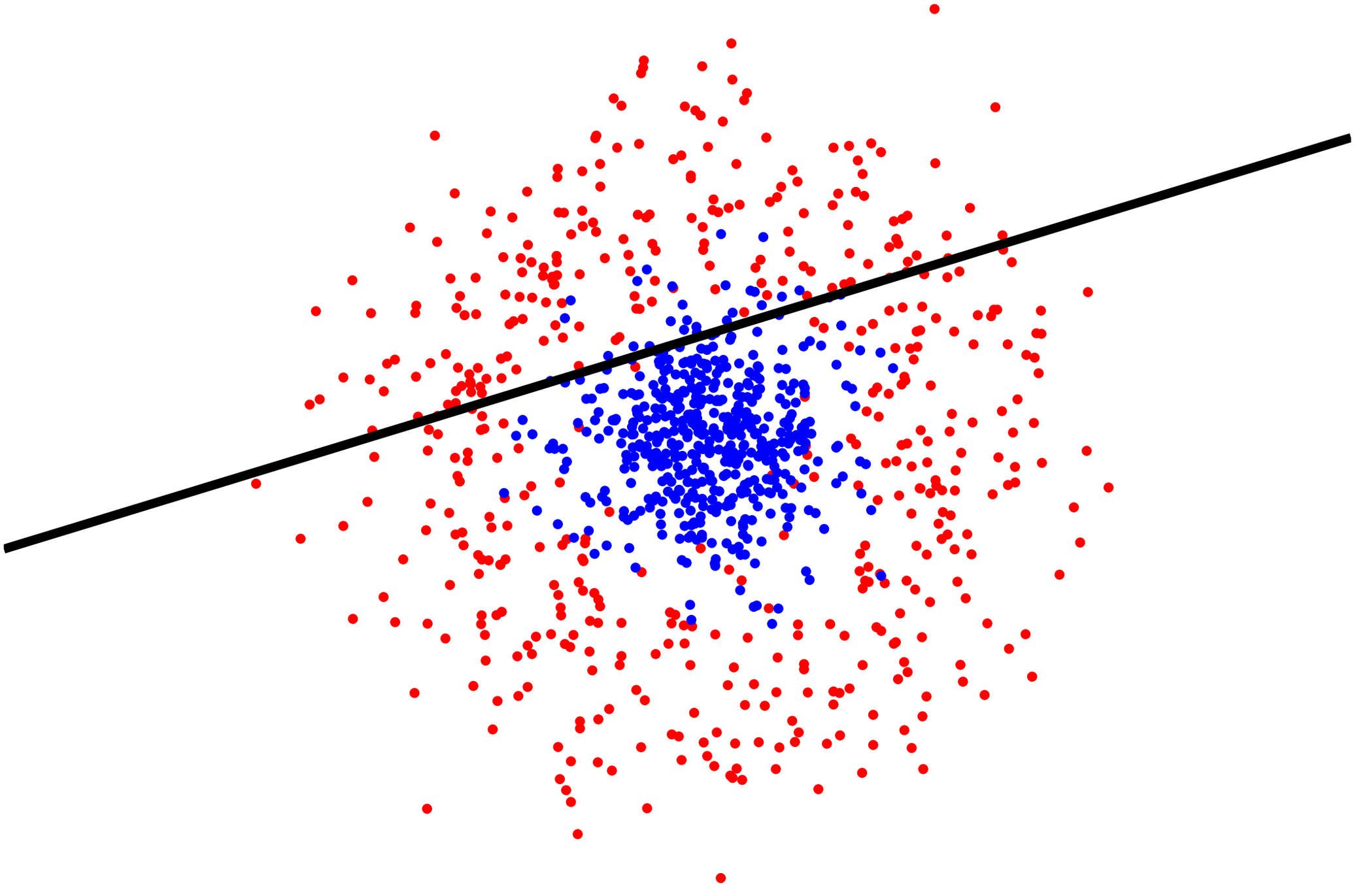


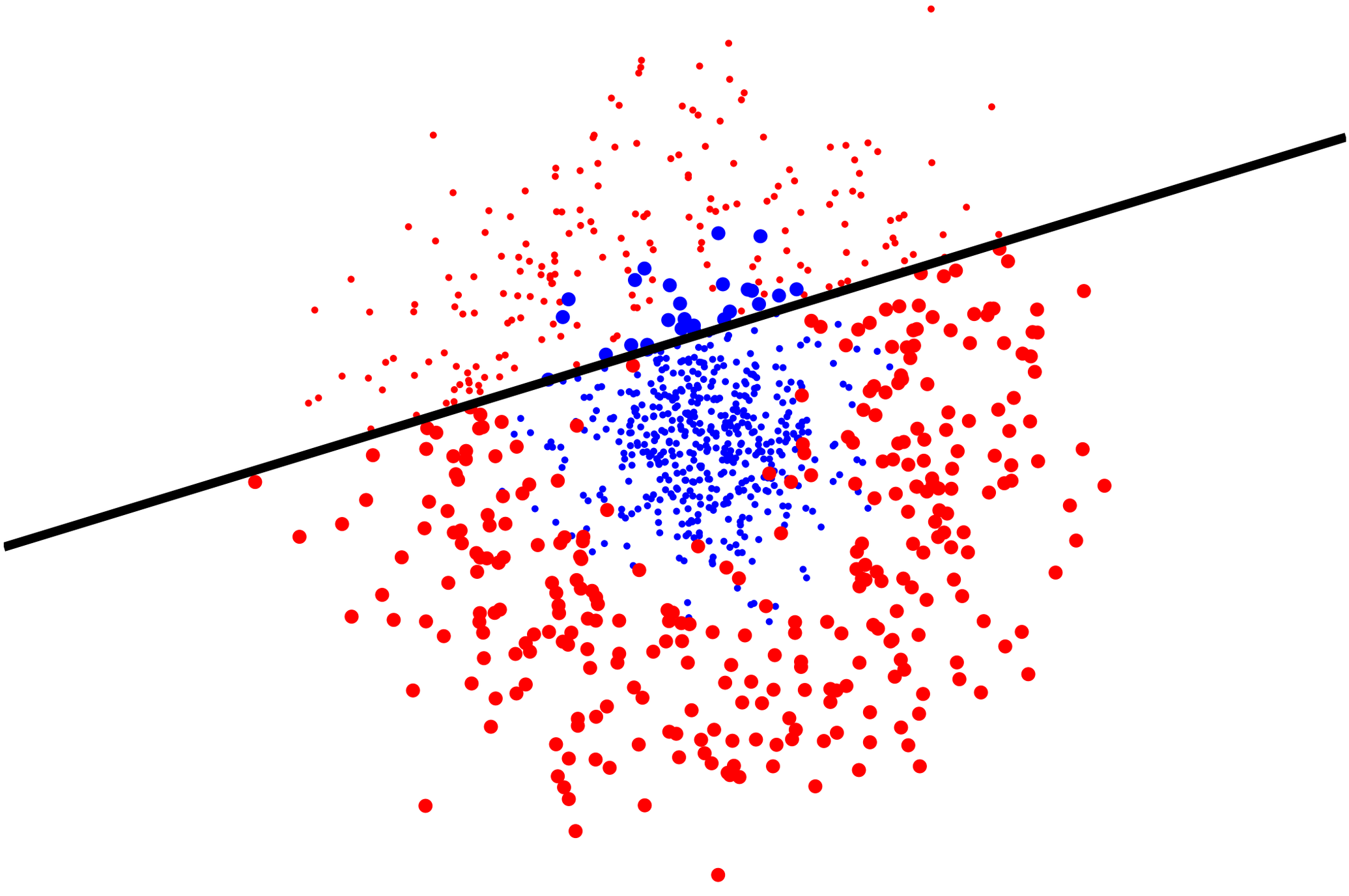


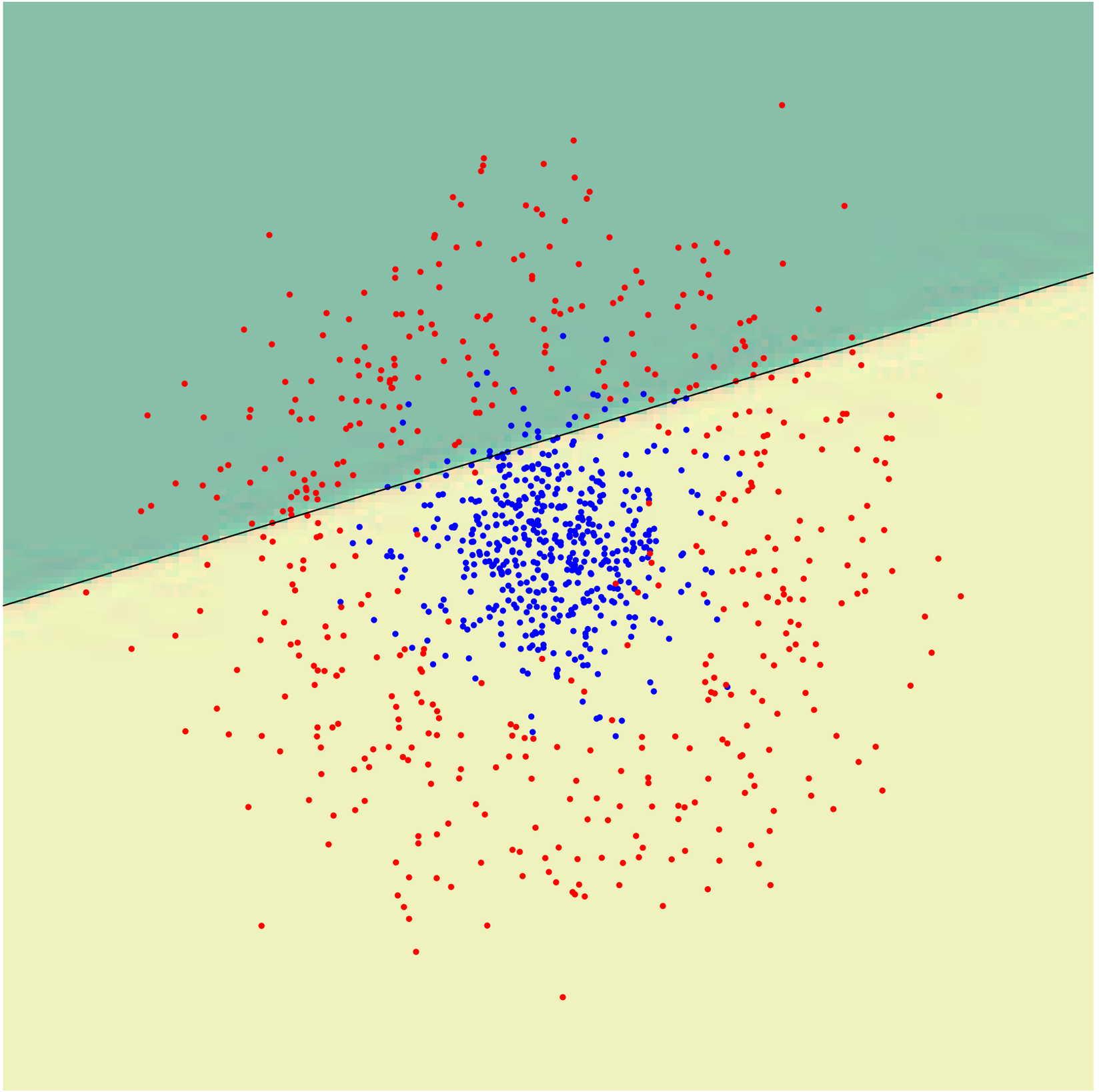


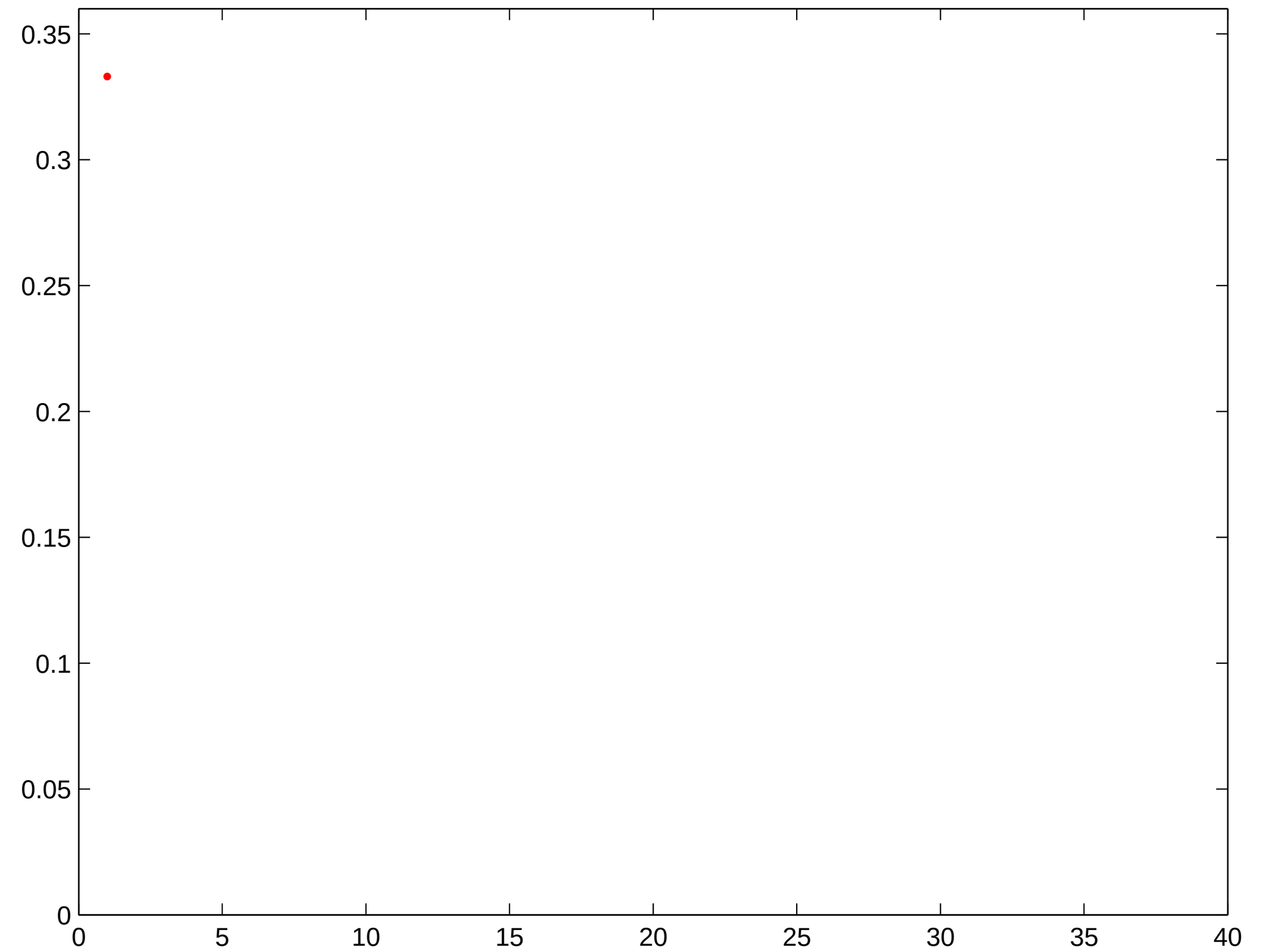


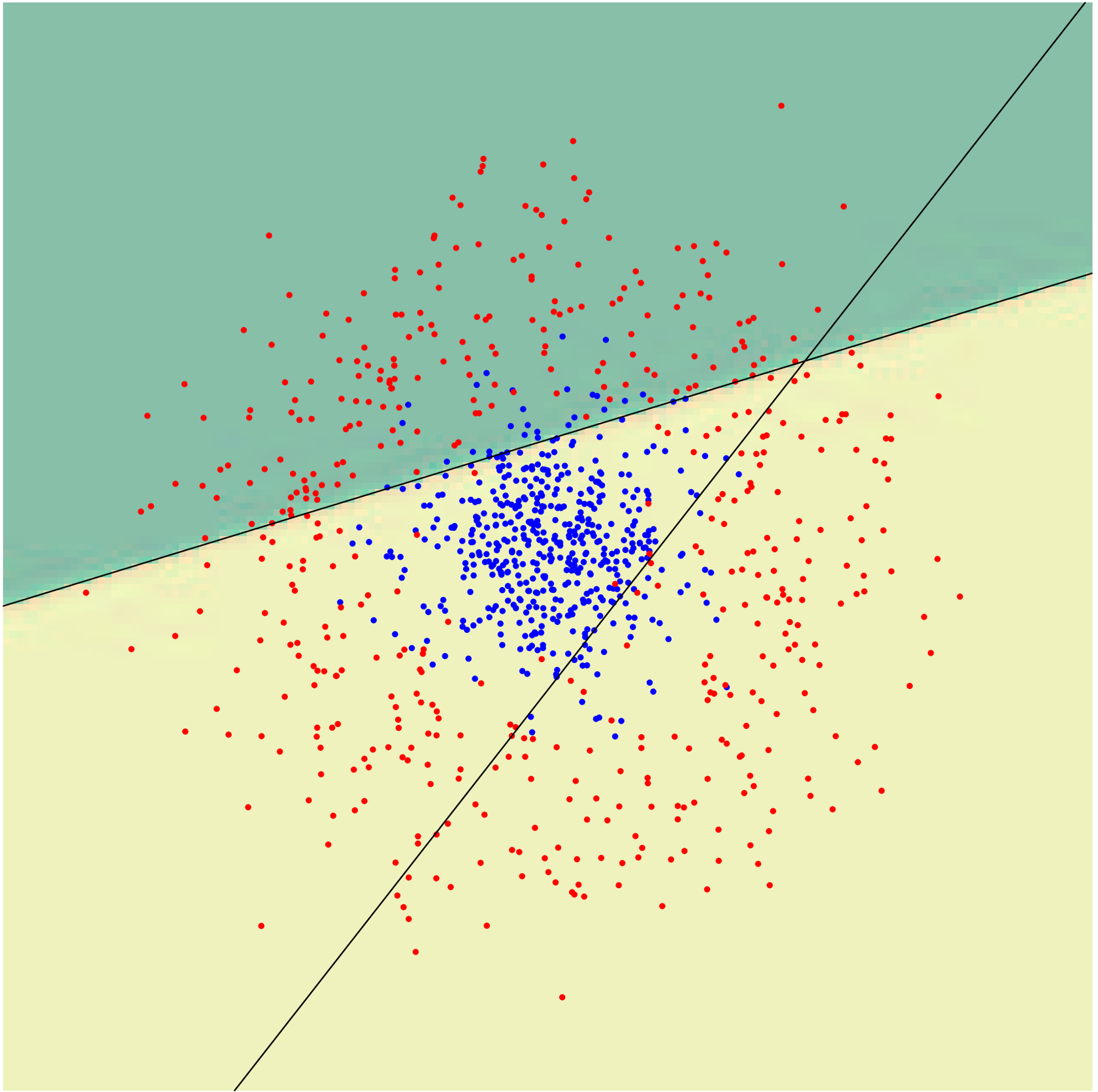


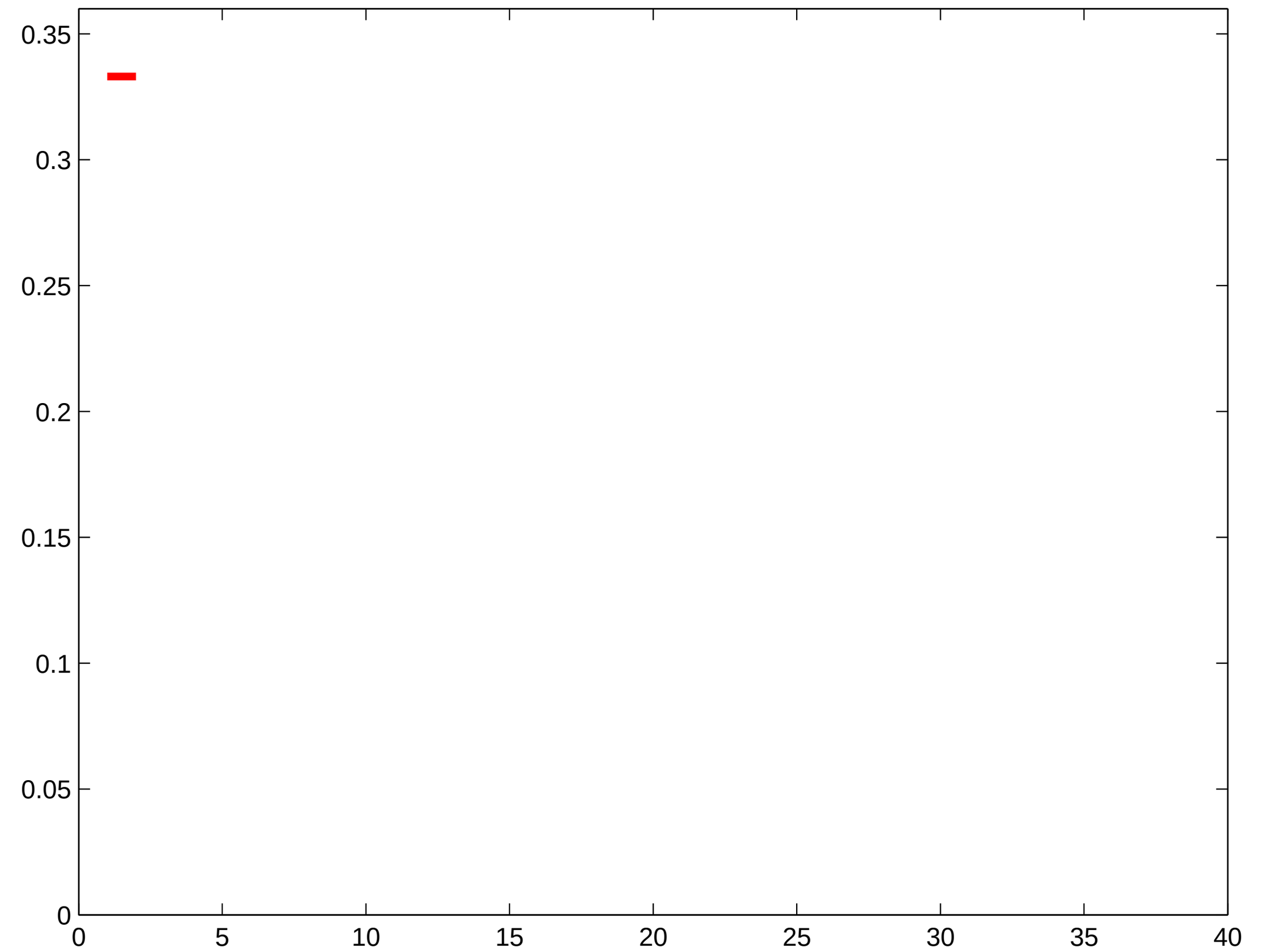


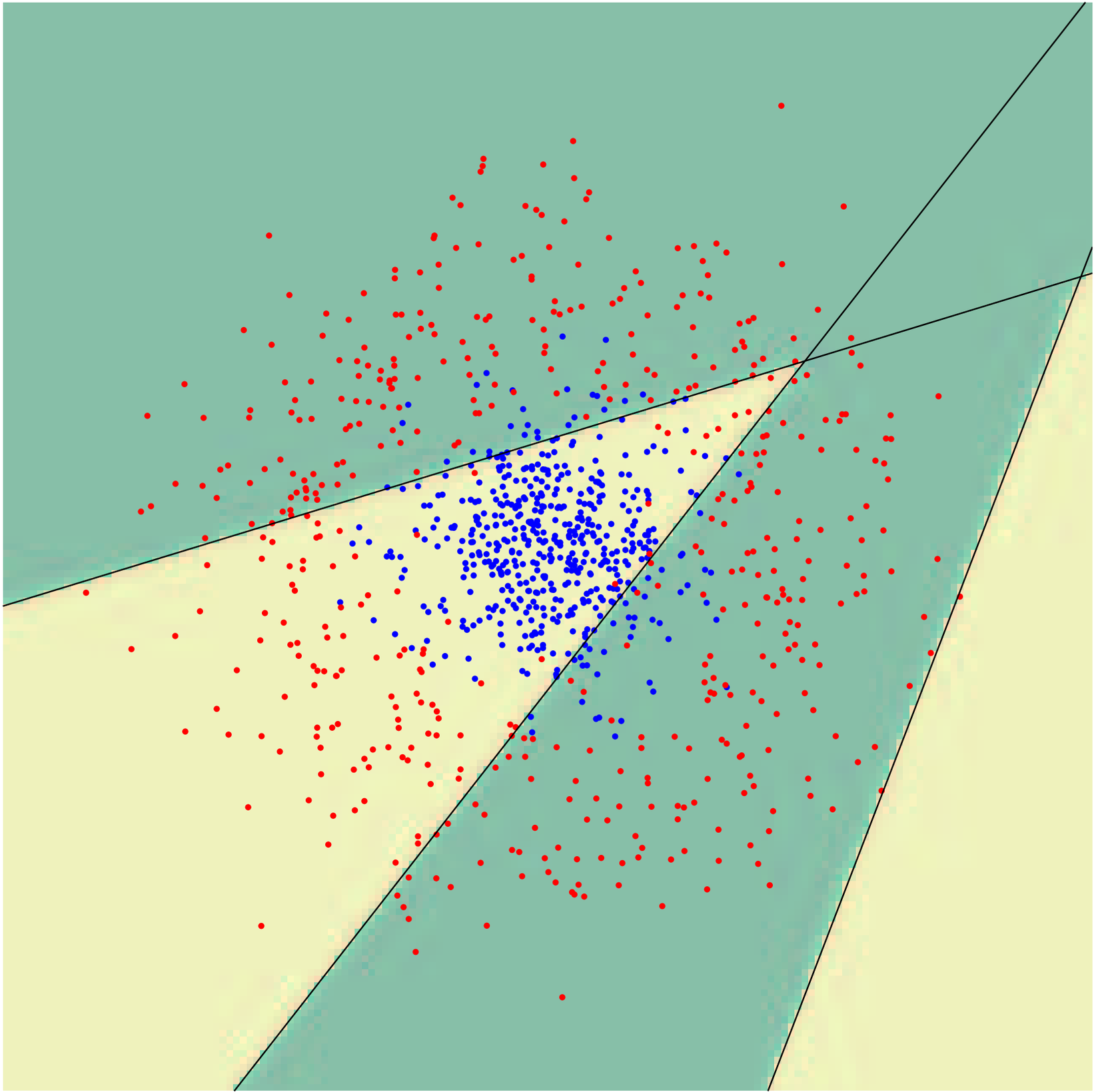


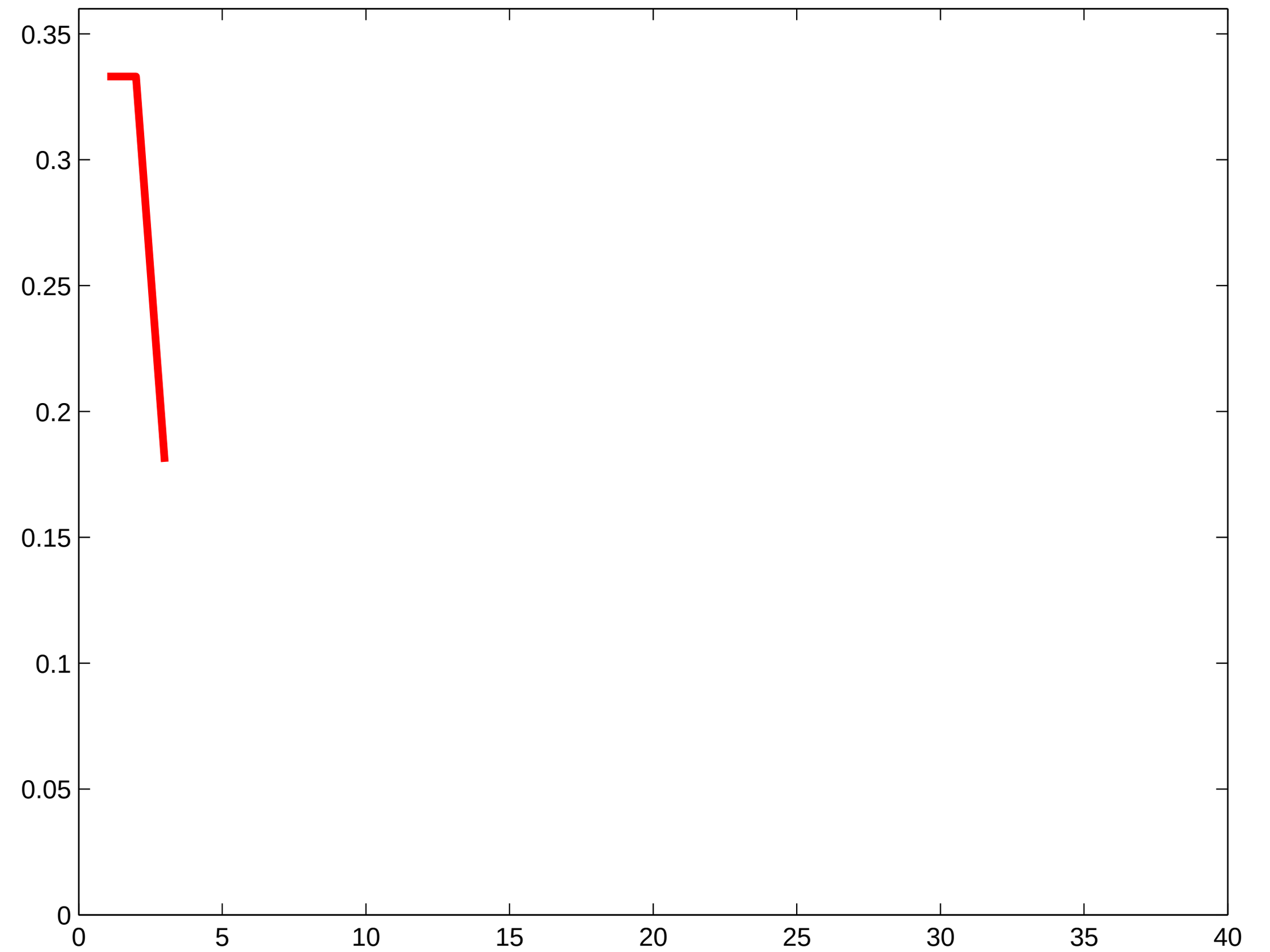


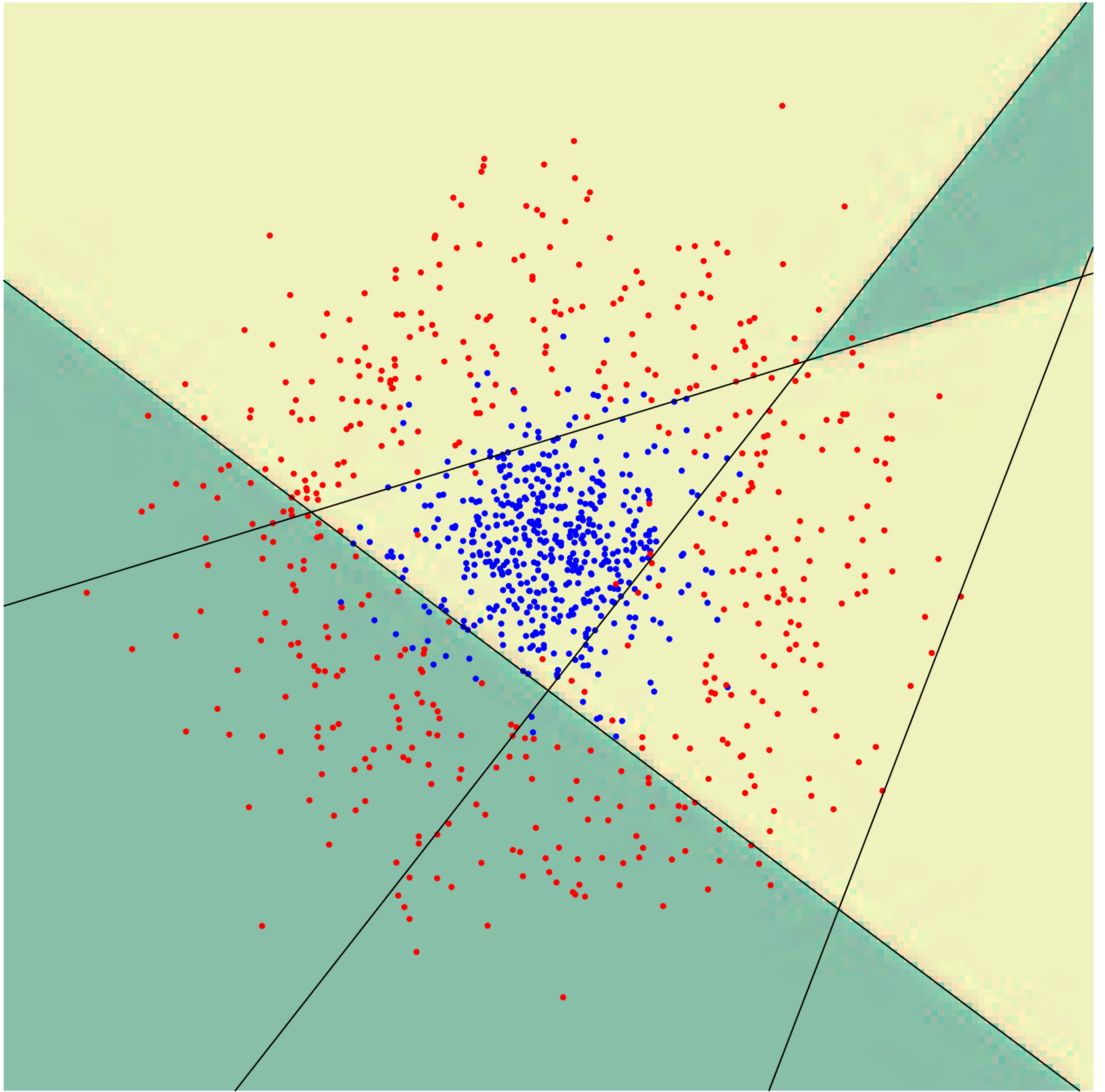

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40

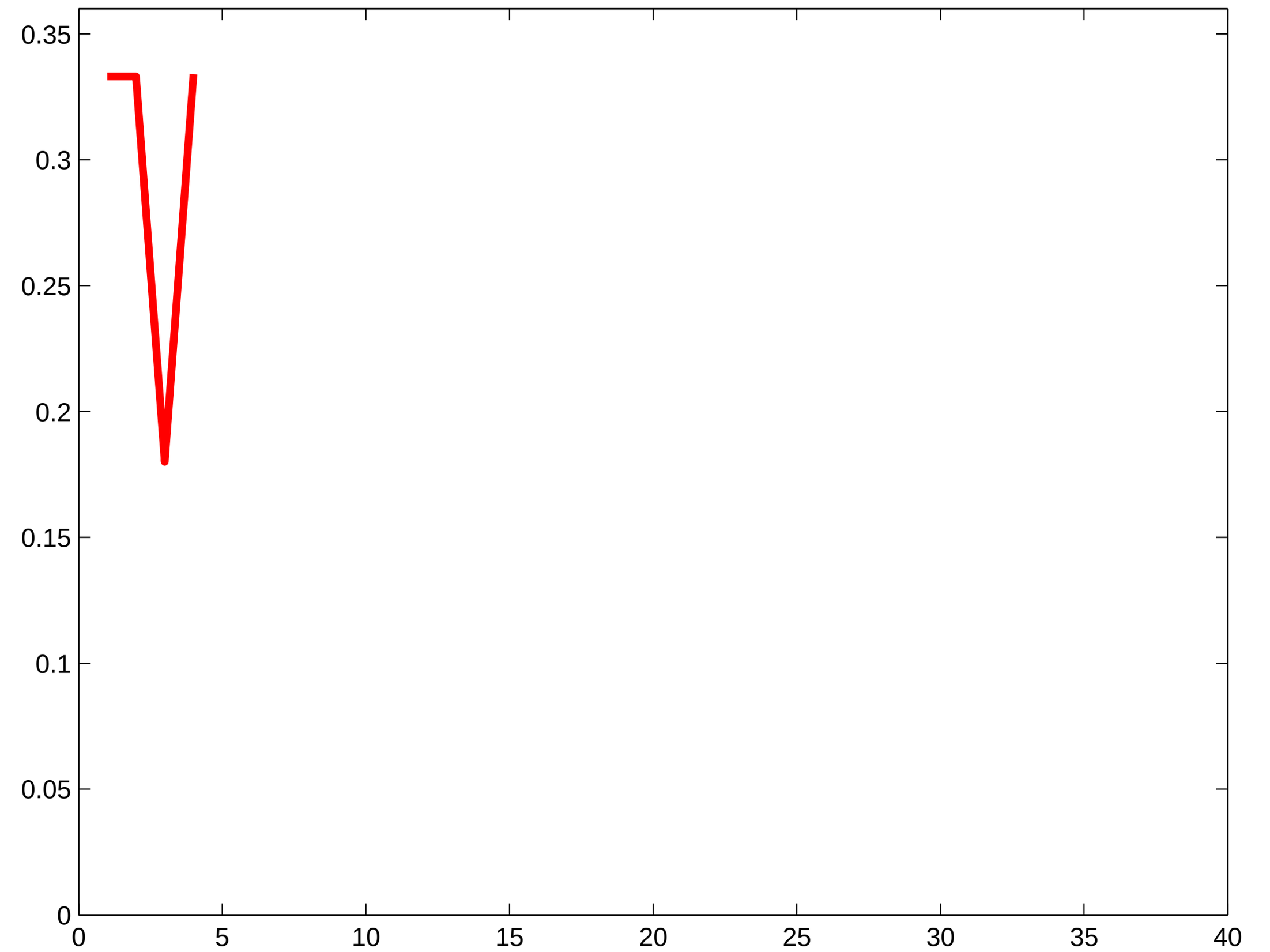


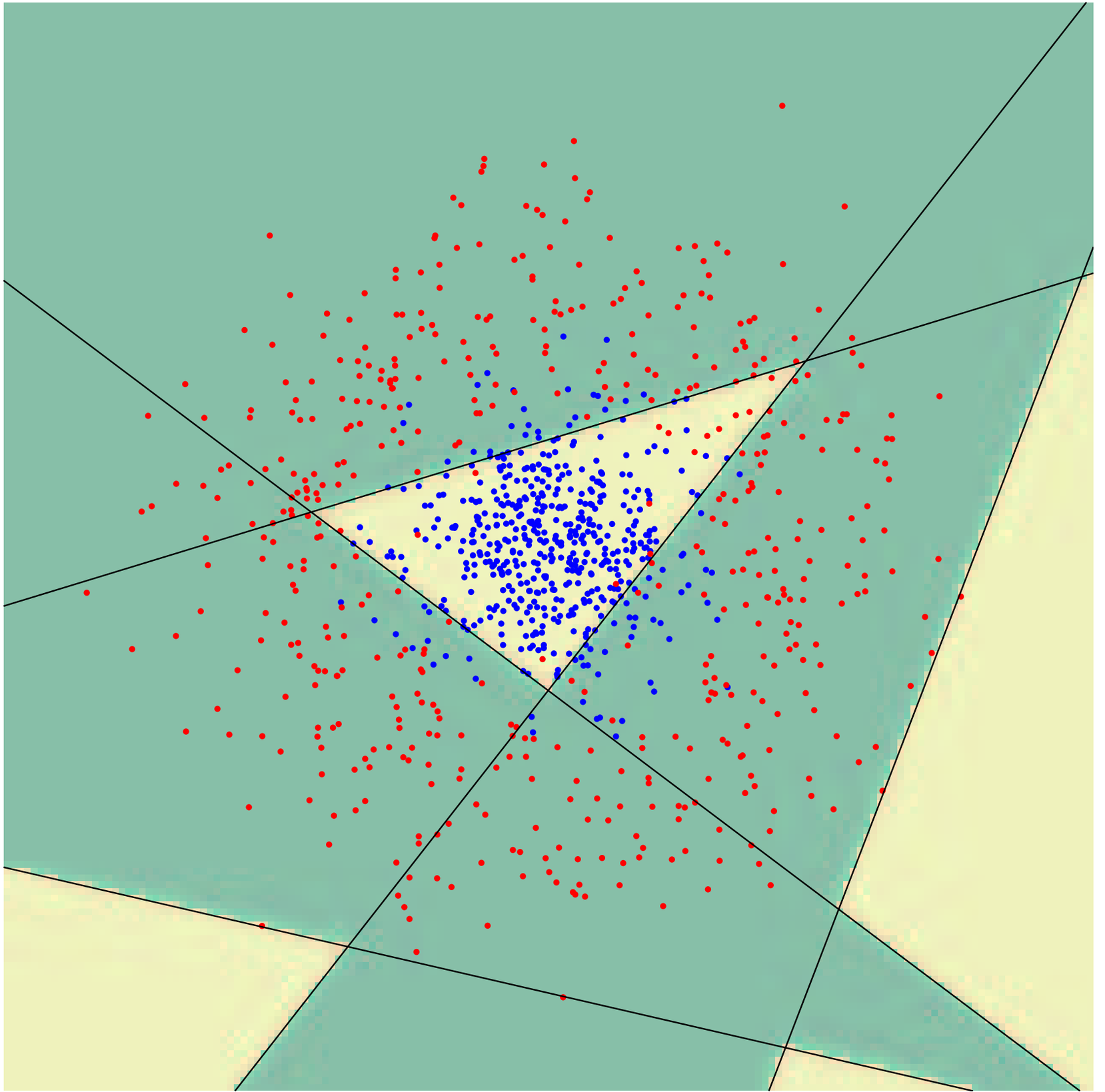

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40

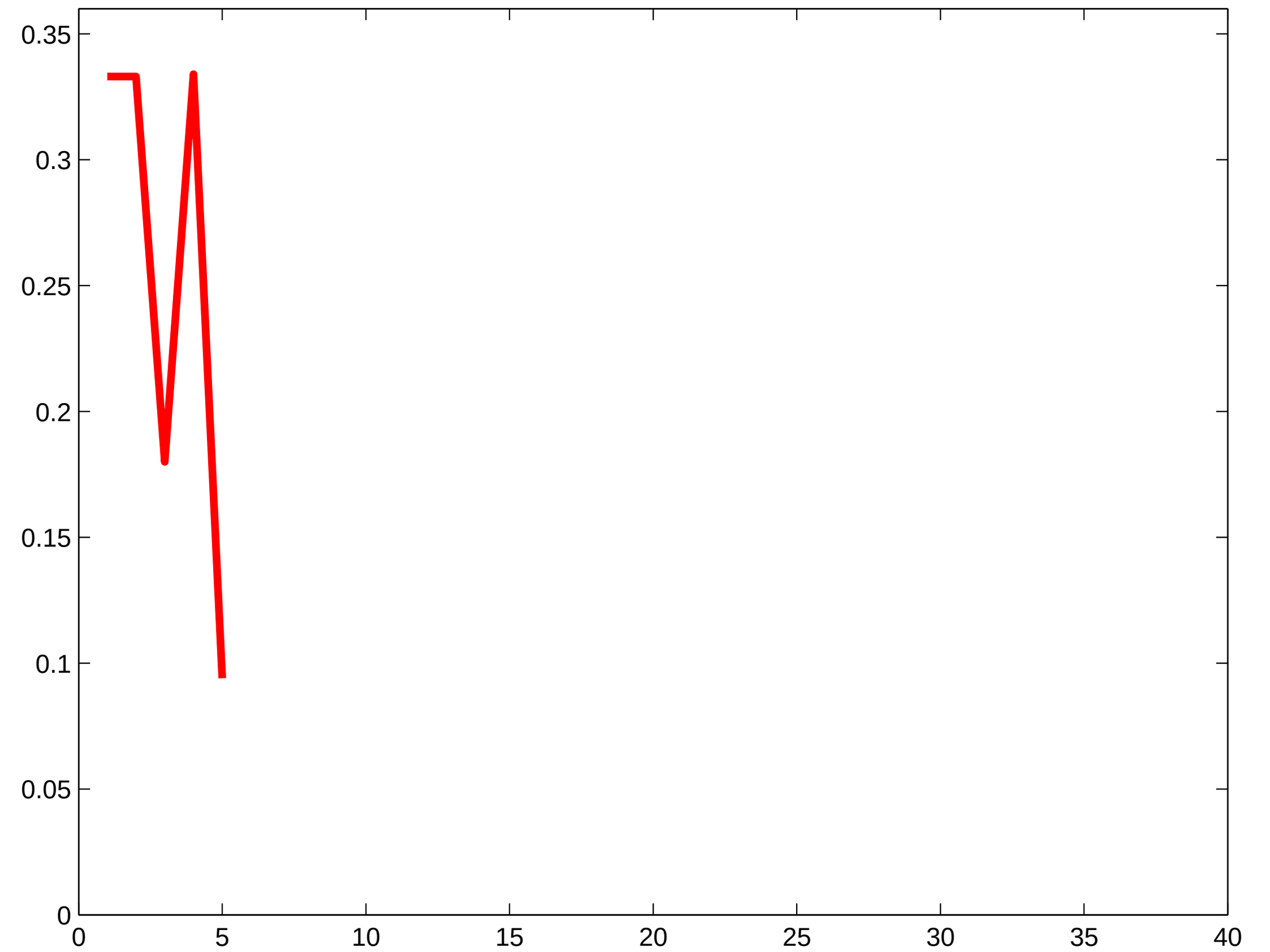


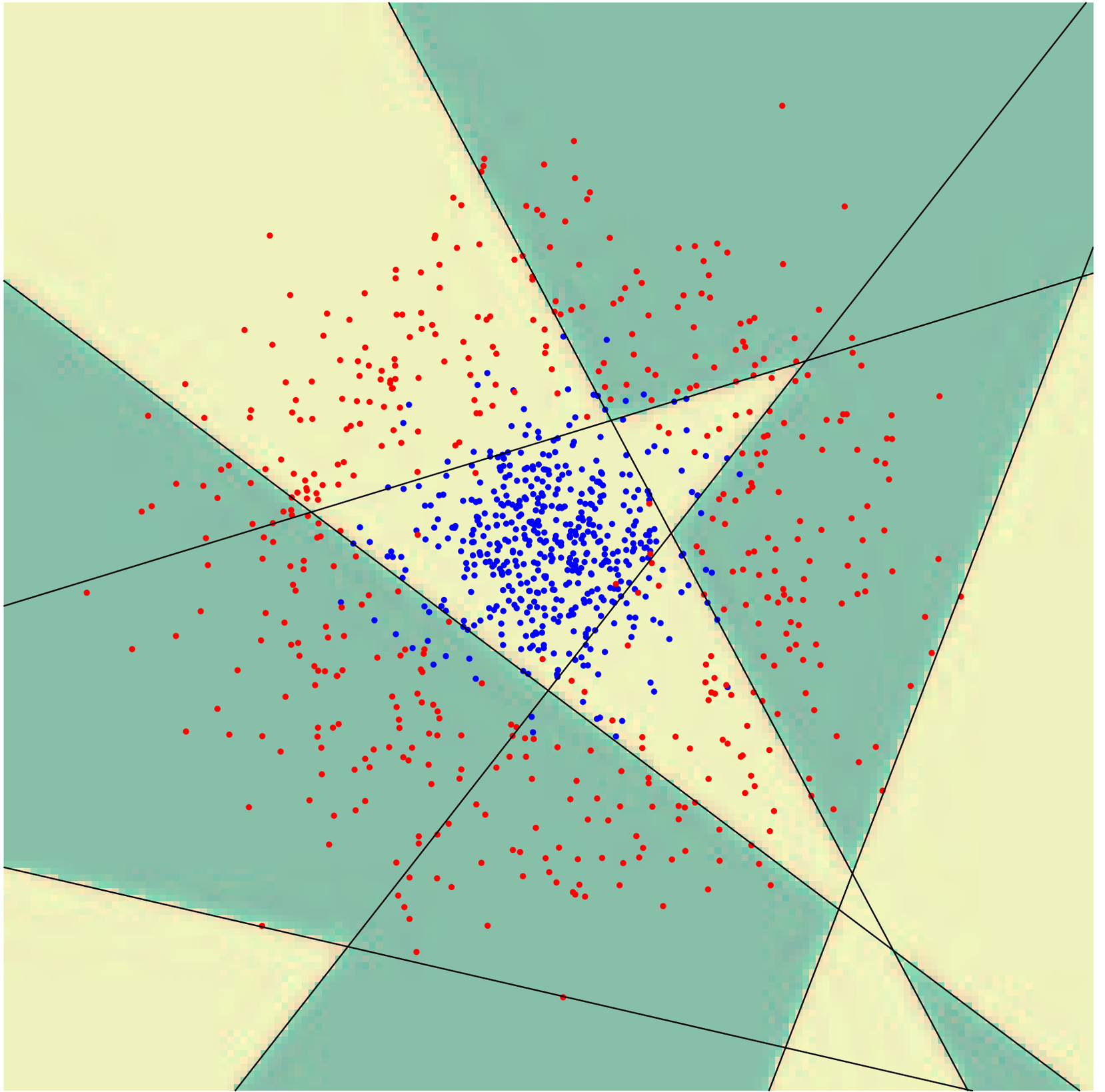

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40

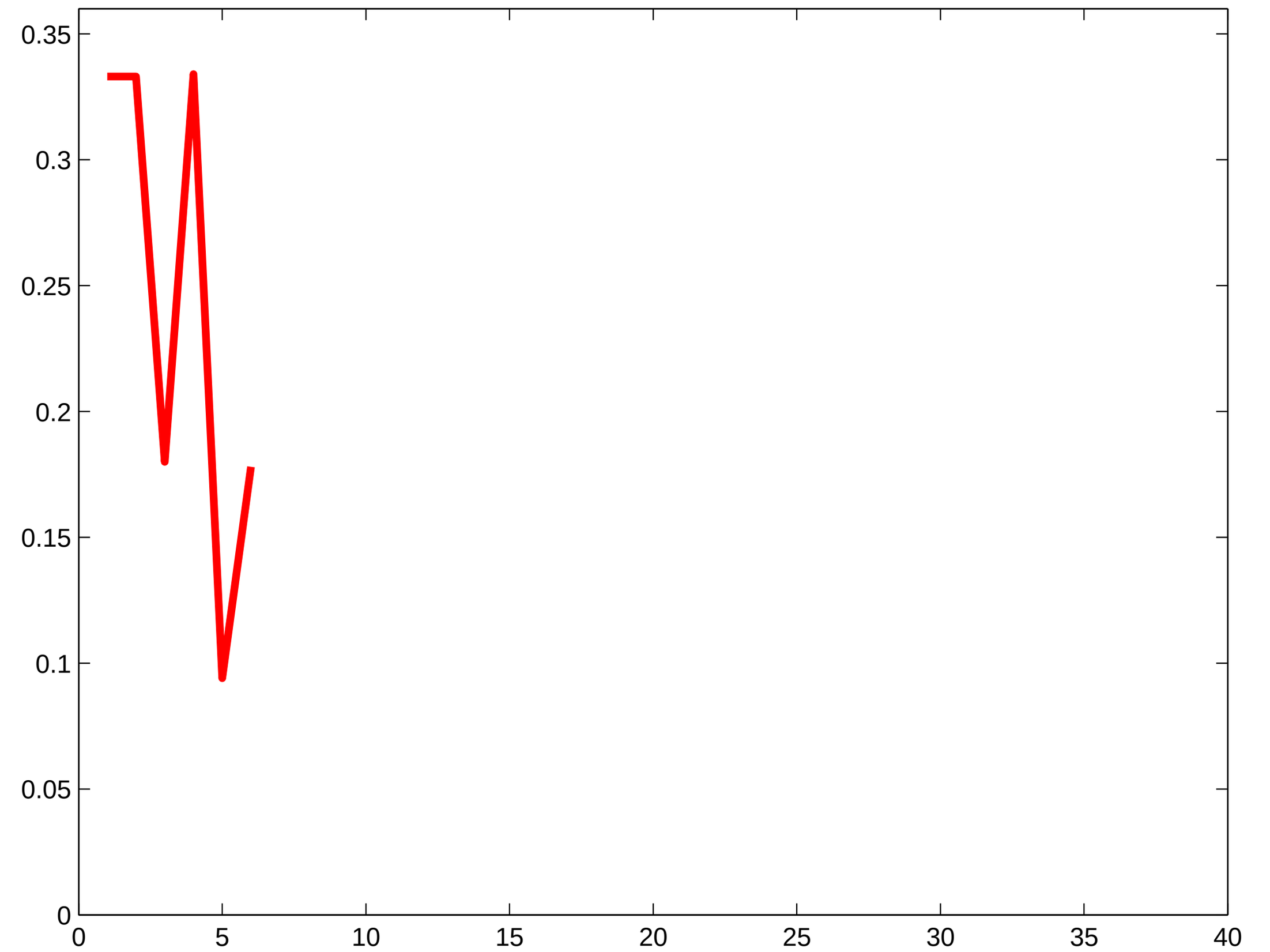


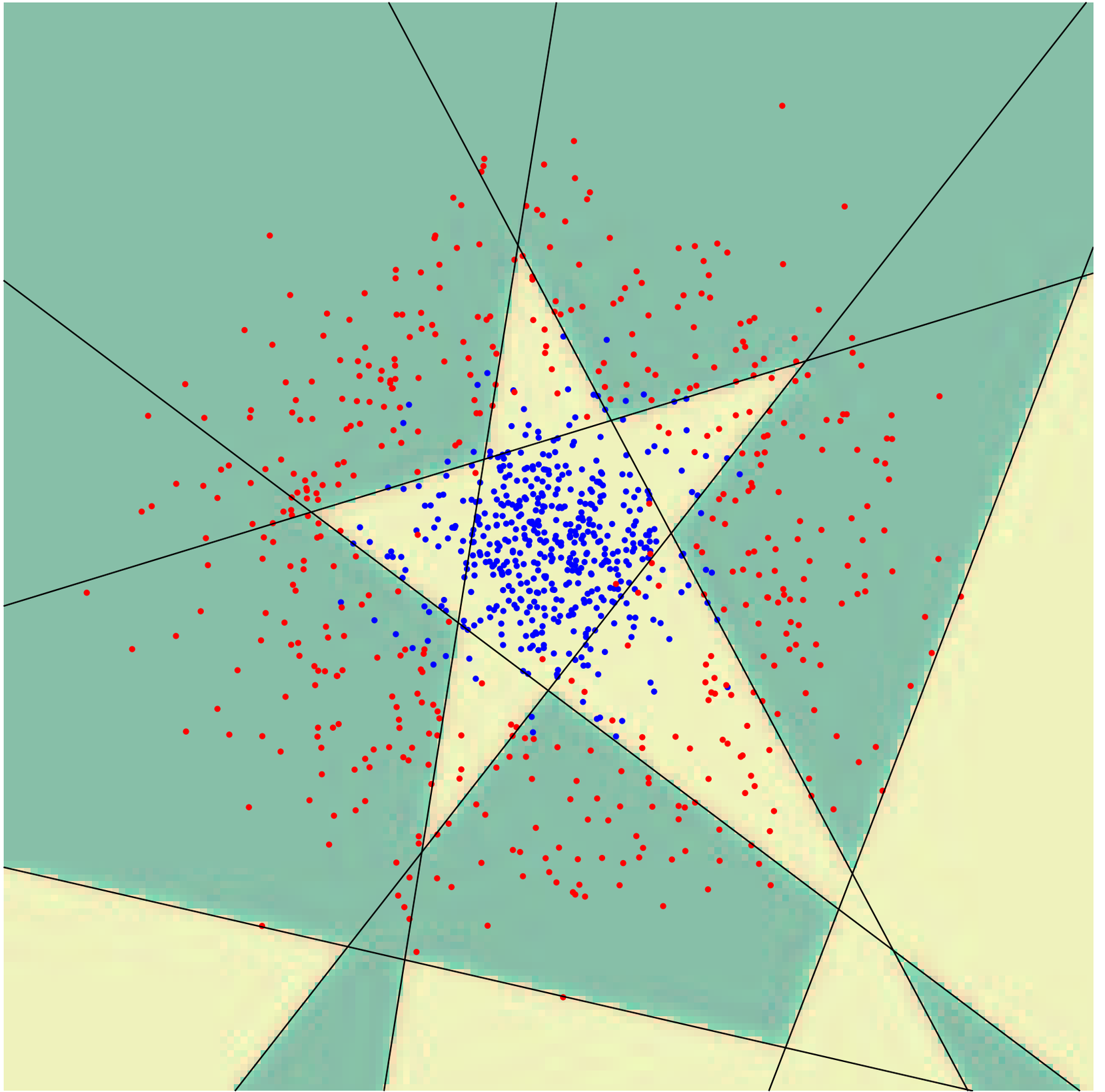

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40

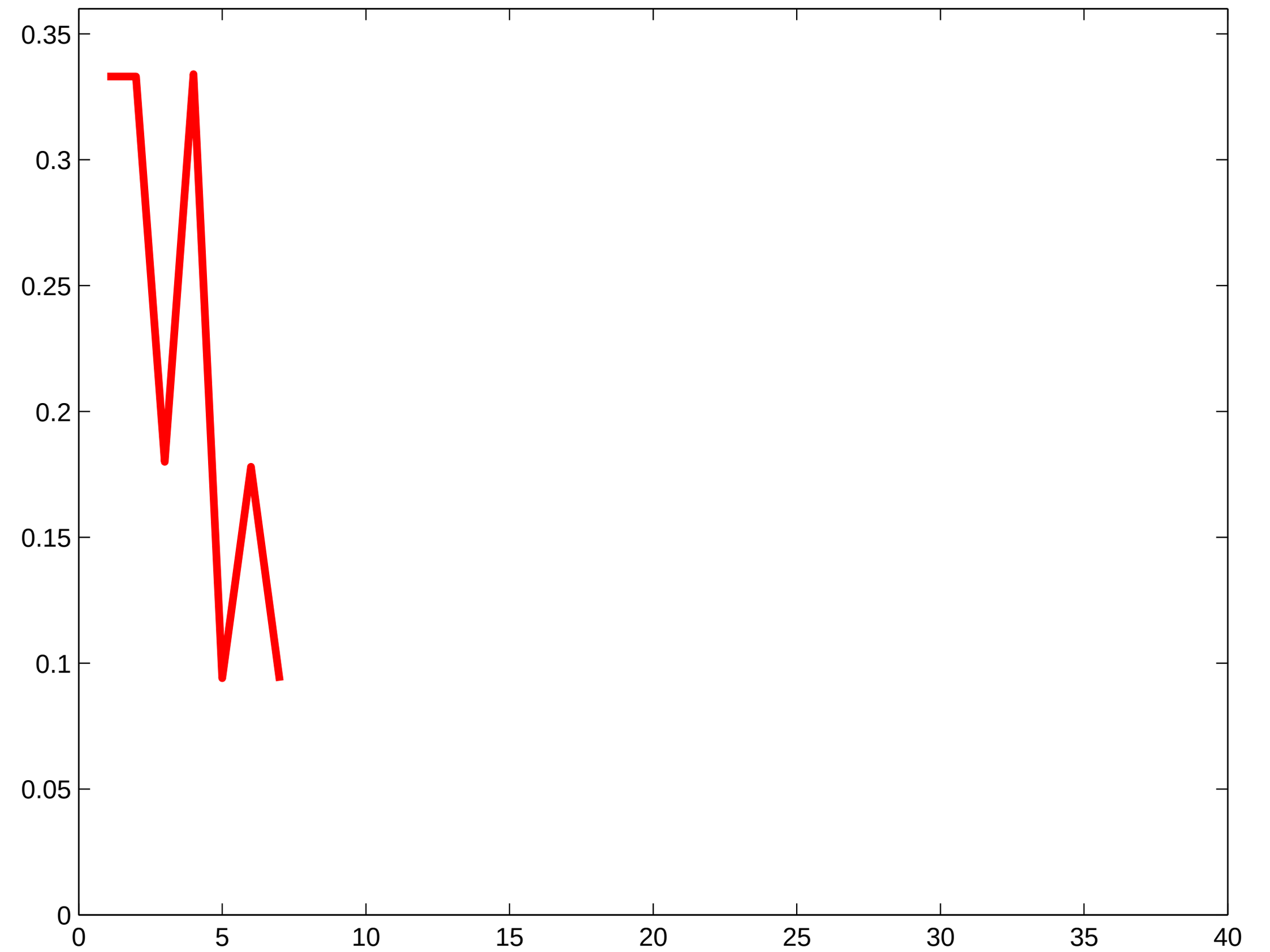


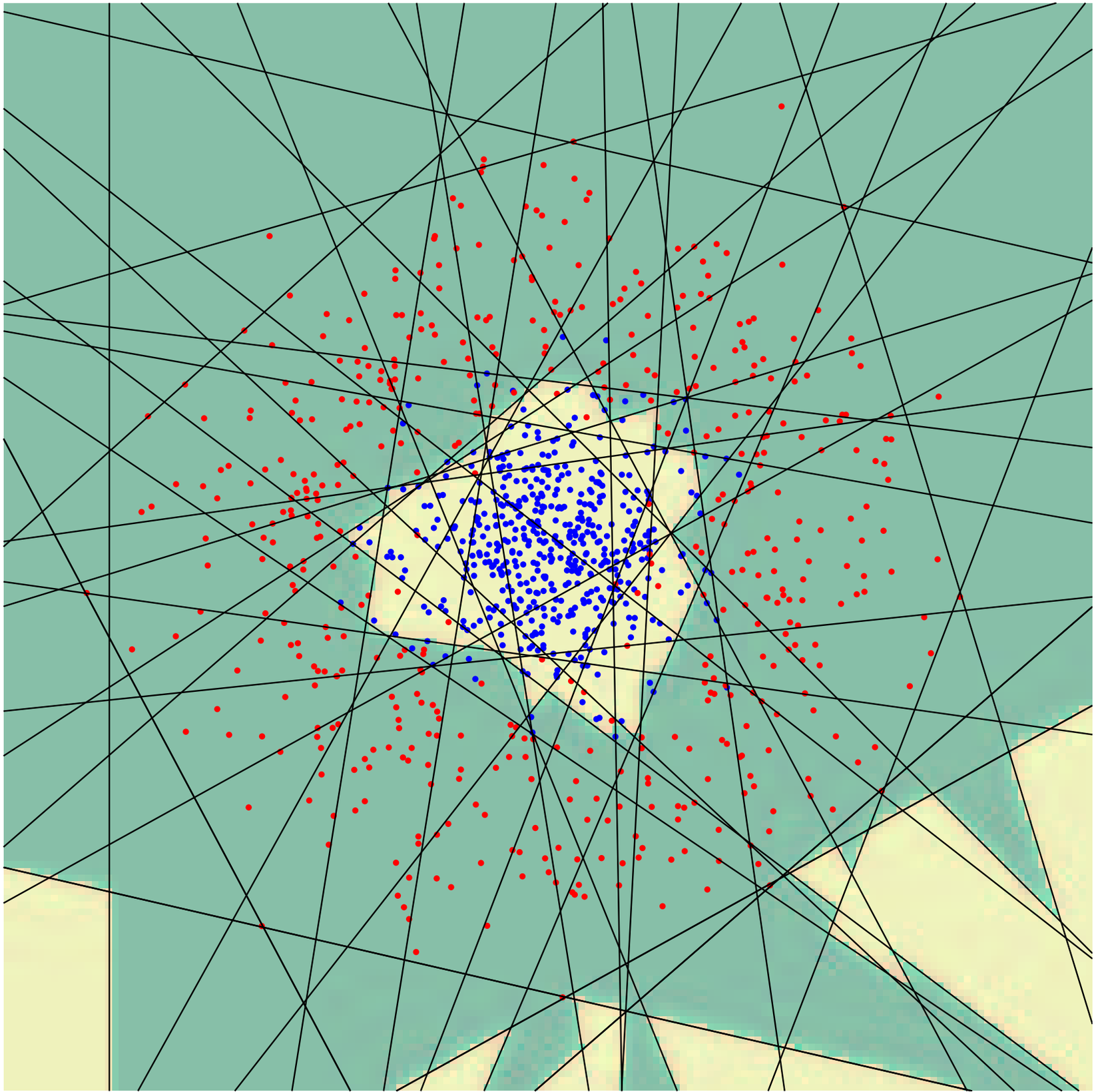

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40

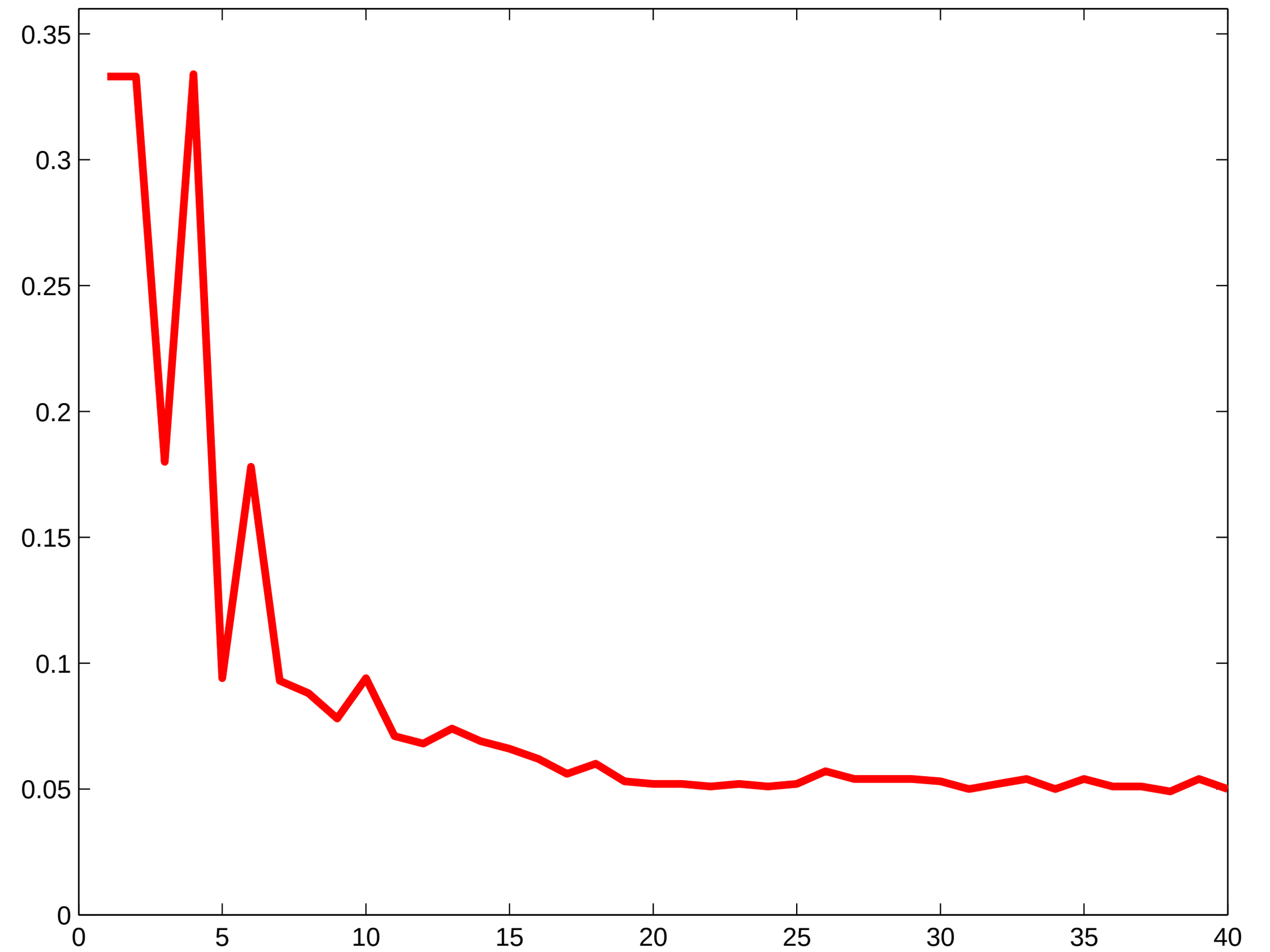


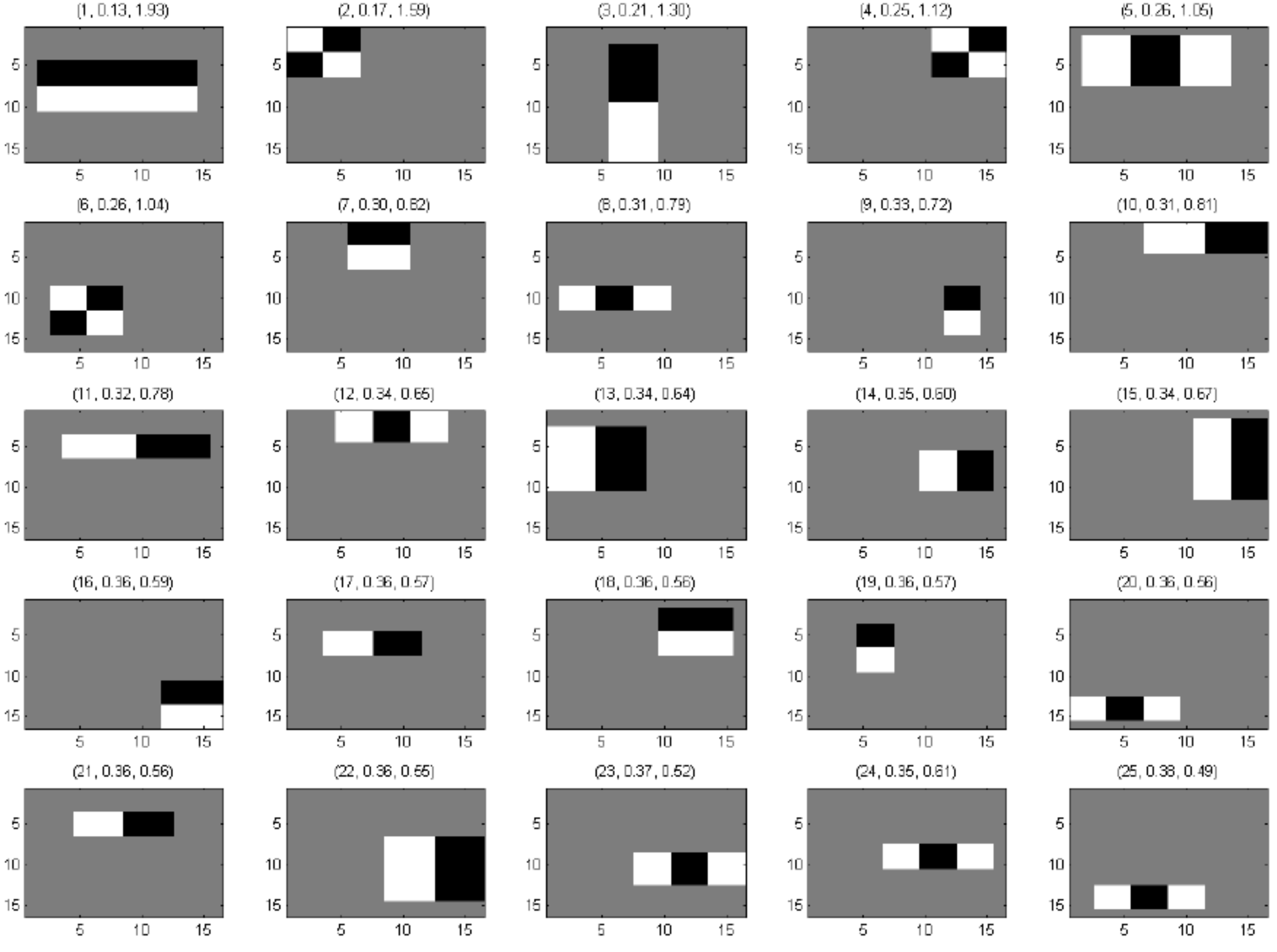

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40




0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40




0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40


(1, 0.13, 1.93)
(2, 0.17, 1.59)
(3, 0.21, 1.30)
(4, 0.25, 1.12)
(5, 0.26, 1.05)
(6, 0.26, 1.04)
(7, 0.30, 0.82)
(8, 0.31, 0.79)
(9, 0.33, 0.72)
(10, 0.31, 0.81)
(11, 0.32, 0.78)
(12, 0.34, 0.65)
(13, 0.34, 0.64)
(14, 0.35, 0.60)
(15, 0.34, 0.67)
(16, 0.36, 0.59)
(17, 0.36, 0.57)
(18, 0.36, 0.56)
(19, 0.36, 0.57)
(20, 0.36, 0.56)
(21, 0.36, 0.56)
(22, 0.36, 0.55)
(23, 0.37, 0.52)
(24, 0.35, 0.61)
(25, 0.38, 0.49)