

Rozpoznávání tváří I

Vojtěch Franc

Katedra kybernetiky, ČVUT FEL Praha



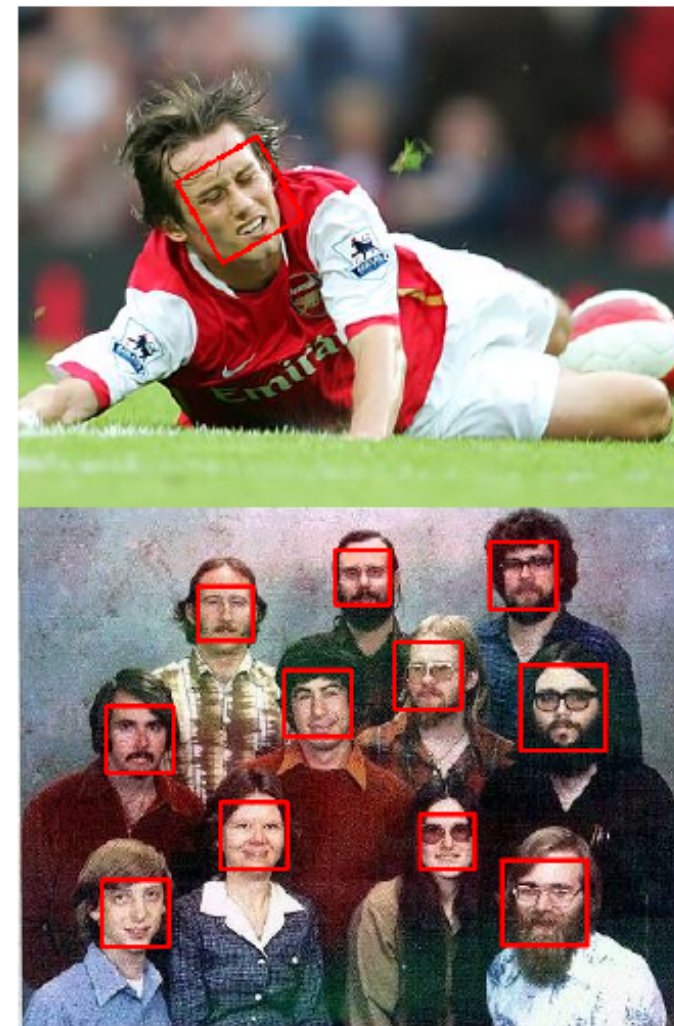
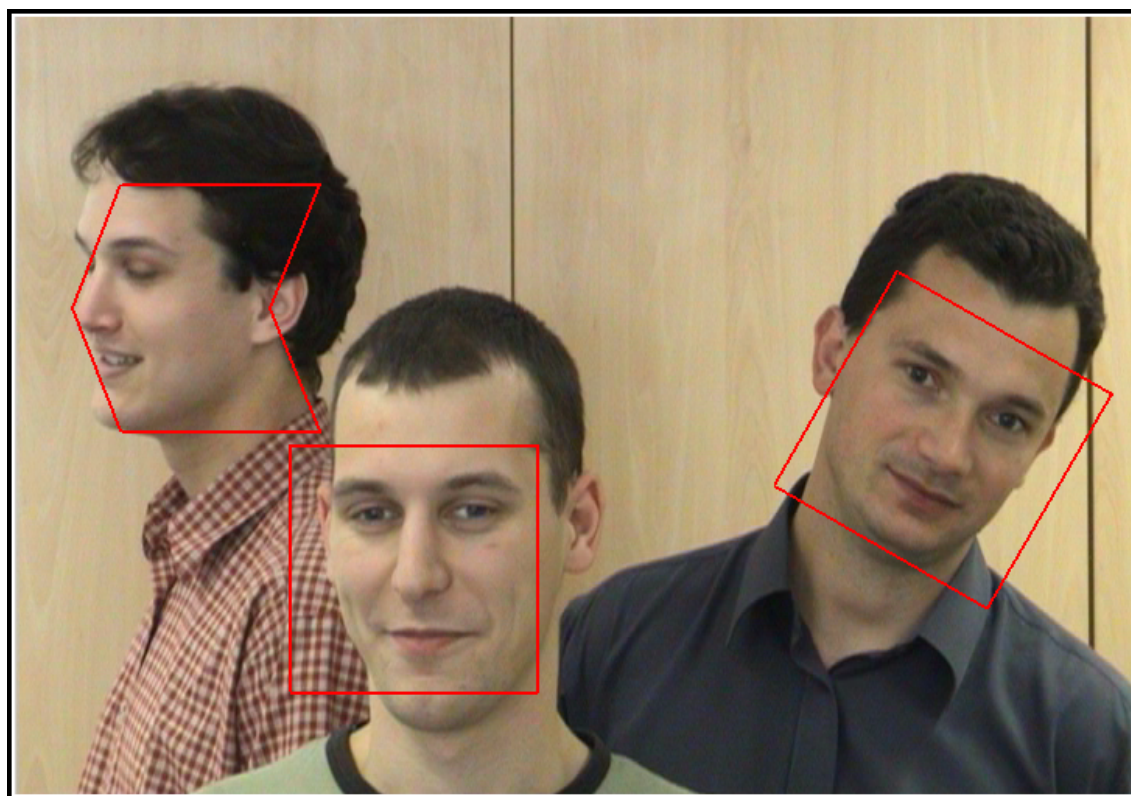
Biometrie ZS 2024

Osnova:

- ◆ Příklady úloh v rozpoznávání tváří: detekce, verifikace, identifikace, odhad věku, ...
- ◆ Metriky pro měření přesnosti
- ◆ Adaboost face detector

Úlohy rozpoznávání tváří: Detekce

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



Metriky pro měření přesnosti detektorů tváří

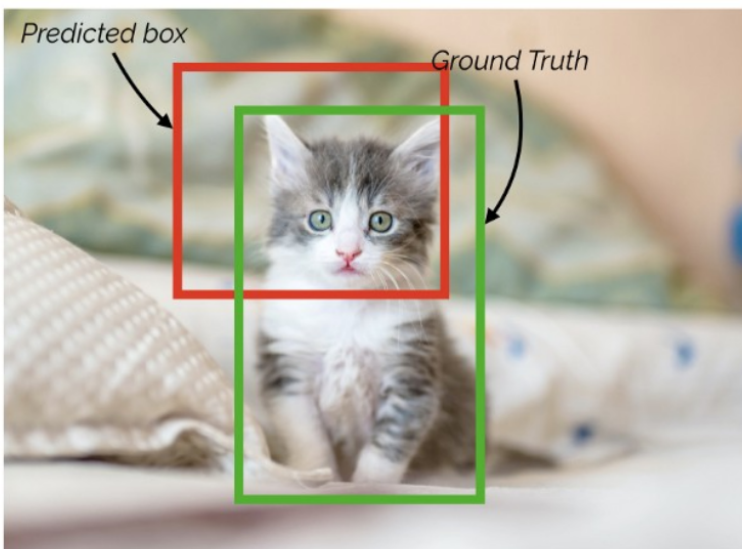
- ◆ Testovací sada: databáze obrázků s manuálně anotovanými pozicemi tváří
- ◆ Správná/falešná detekce definovaná pomocí “Intersection Over Union ratio”

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

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

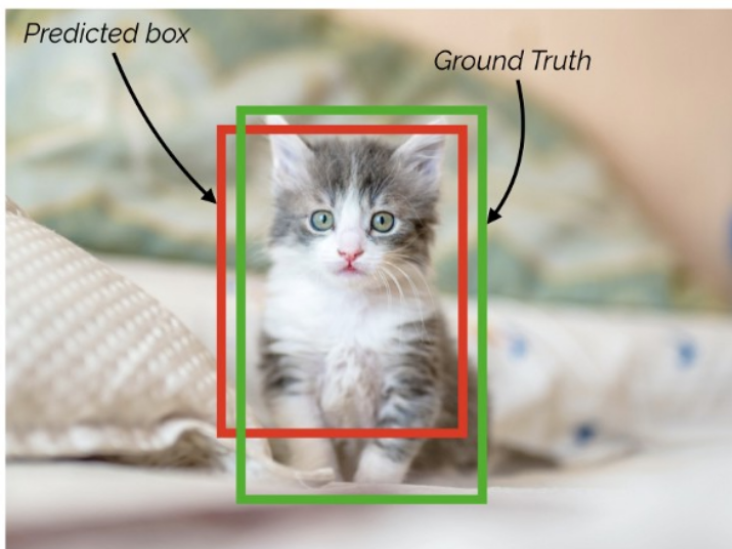
- ◆ $\text{IOU}(A, B) \geq \theta$ (např. $\theta = 0.7$) implikuje správnou detekci a $\text{IOU}(A, B) < \theta$ falešnou detekci.

False Positive (FP)



$\text{IoU} = \sim 0.3$

True Positive (TP)



$\text{IoU} = \sim 0.7$

Měření přesnosti detektorů tváří pomocí ROC křivky

- ◆ True Positive Rate, Sensitivita, Recall (stejné názvy pro to samé)

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

Podíl tváří, které se podaří vytěžit z databáze.

- ◆ False Positive Rate per image

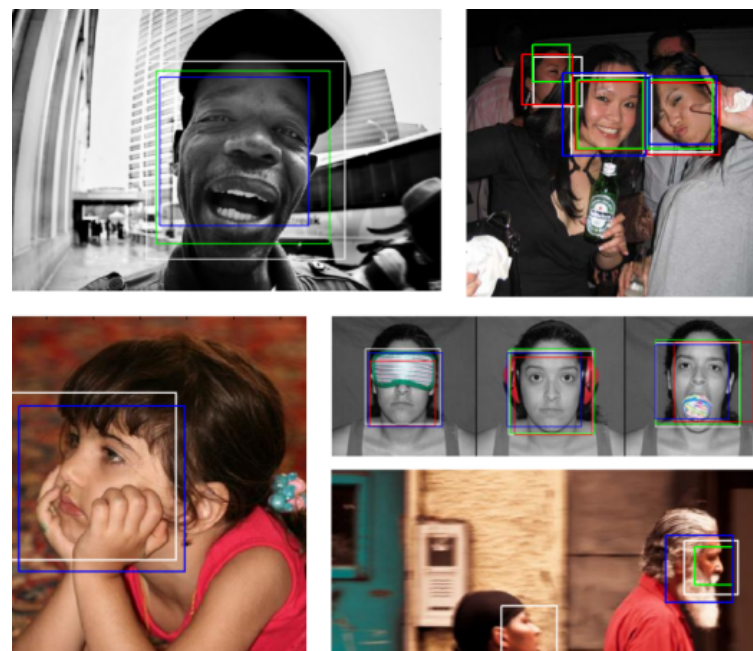
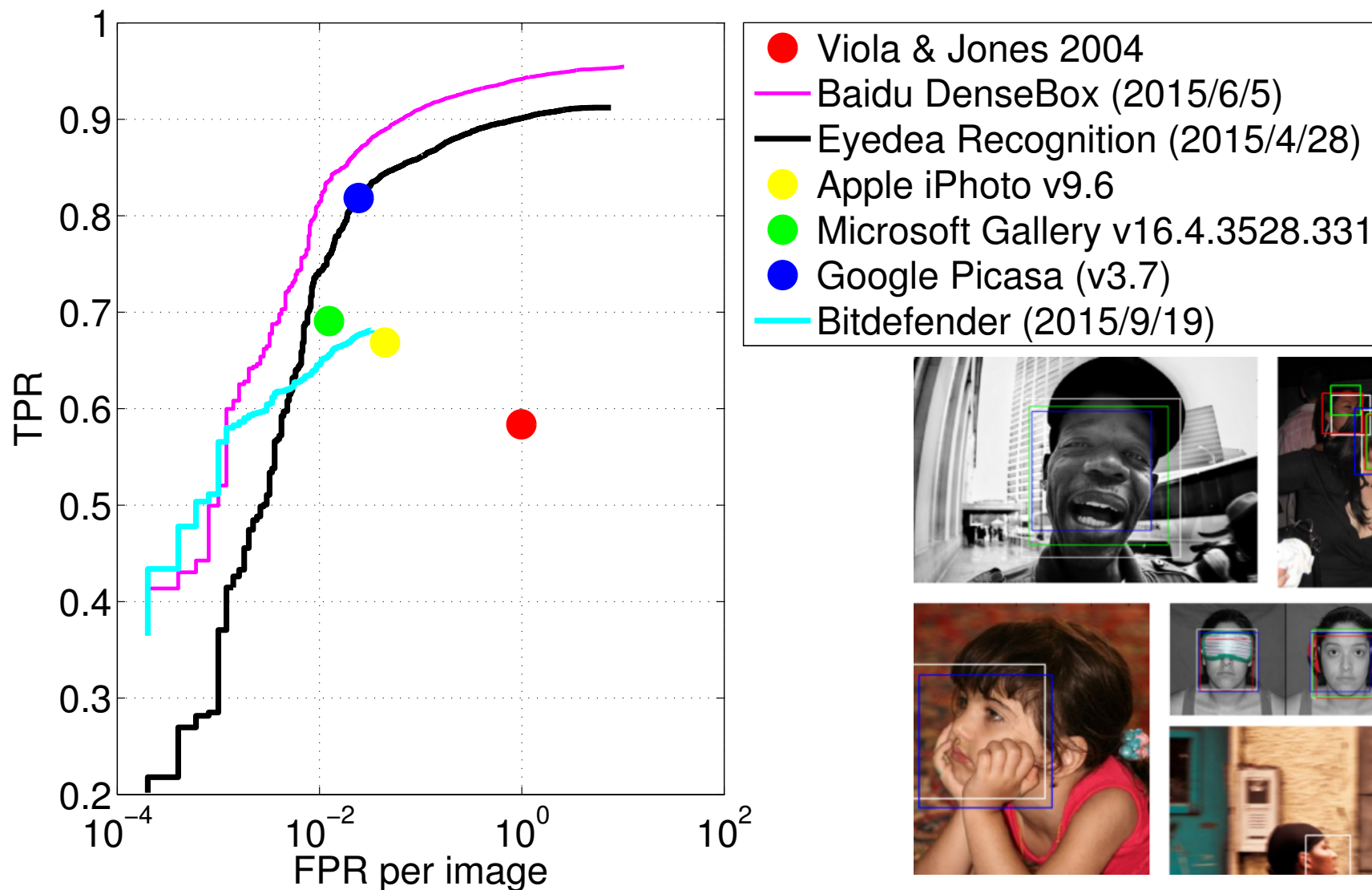
$$\text{FPR per image} = \frac{\text{počet falešných detekcí}}{\text{počet všech obrázků}} \in [0, \infty)$$

Průměrný počet chybných detekcí v obrázku

- ◆ ROC křivka ukazuje hodnoty TPR a FPR jako funkci rozhodovacího prahu (rozhodovací práh je nastavitelný parametr, který určuje citlivost detektoru).

Přesnost detektorů na reálných datech

- ◆ Fine-grained Evaluation on Face Detection in The Wild [Yang et al 2015] je benchmark pro měření přesnosti detektorů tváří.
- ◆ Multi-Attribute Labelled Faces (MALF) 5,250 obrázků s 11,931 tvářemi.



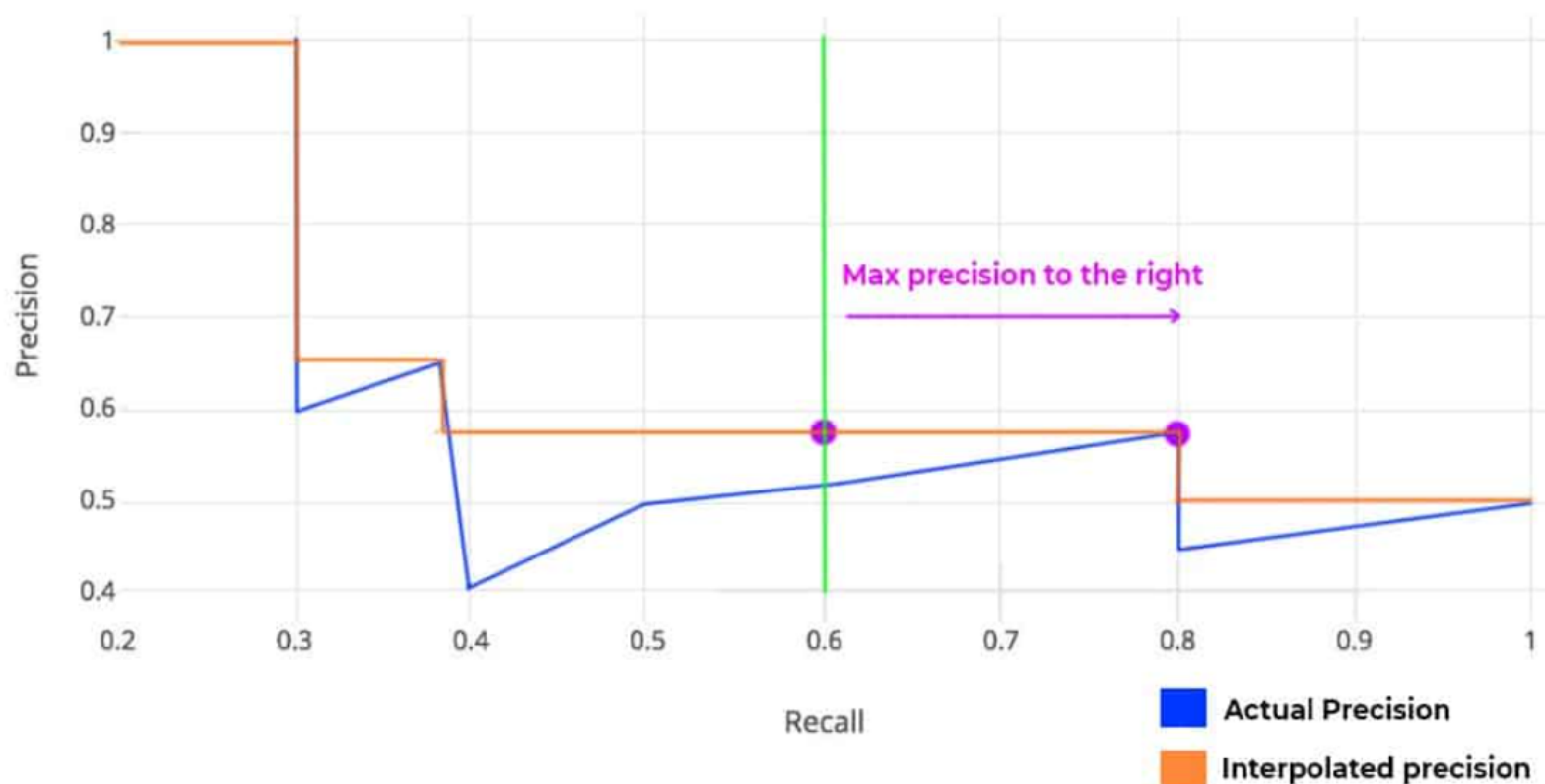
Měření přesnosti detektorů tváří pomocí Precision-Recall křivky

◆ Precision:

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

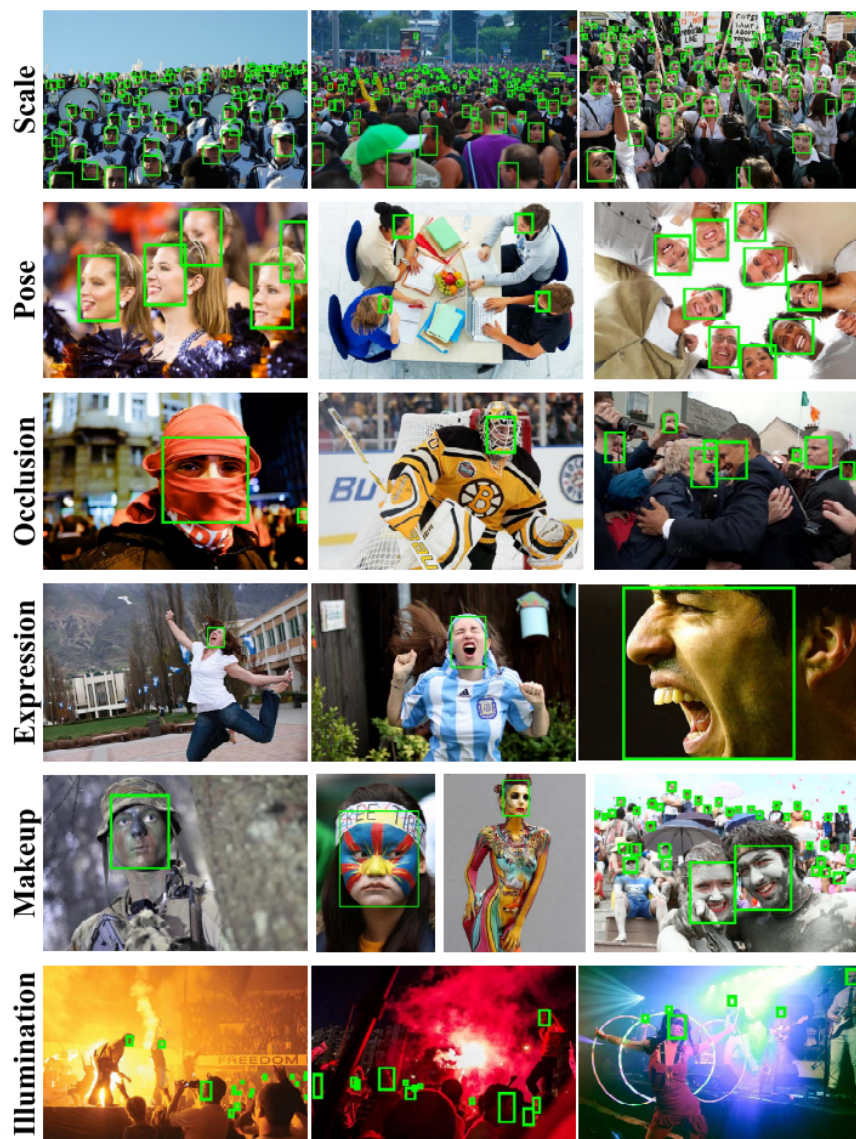
Jaký podíl detekovaných tváří tvoří správné detekce.

- ◆ Precision-Recall křivka ukazuje hodnotu Precision a Recall (TPR) jako funkci rozhodovacího prahu.
- ◆ Average Precision (AP) je aproximace plochy pod Precision-Recall křivkou.

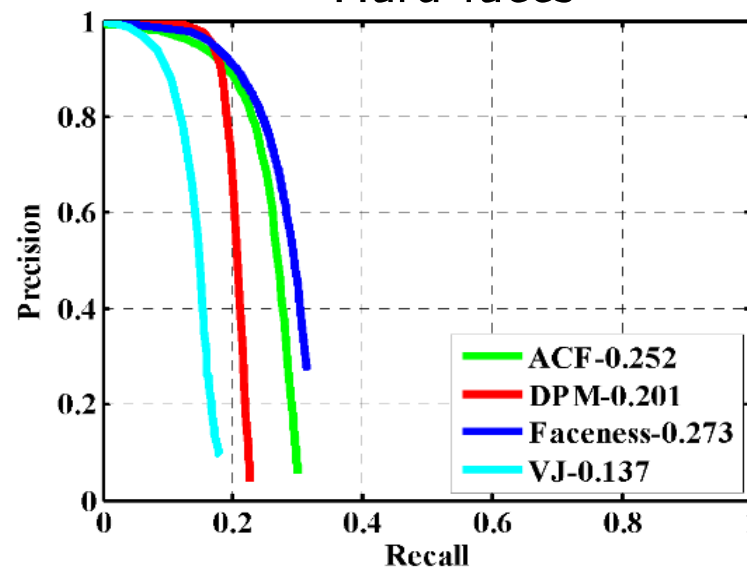


Přesnost detektorů na reálných datech

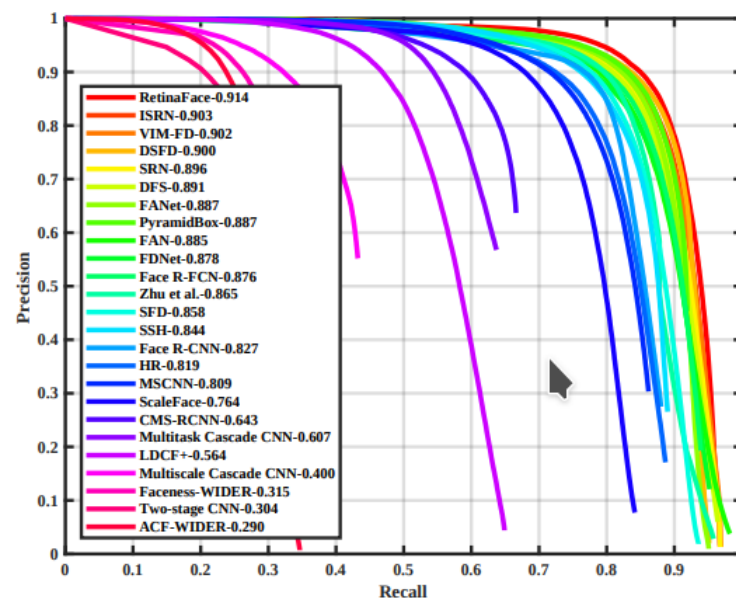
- ♦ WIDER FACE: A Face detection benchmark [Yang 2016]. 32,203 obrázků s 393,703 anotovanými tvářemi s vysokou variabilitou v měřítku, natočení a zákrytech.
- ♦ Databáze je rozdělena tři úrovně složitosti: easy, medium a hard.



Hard faces



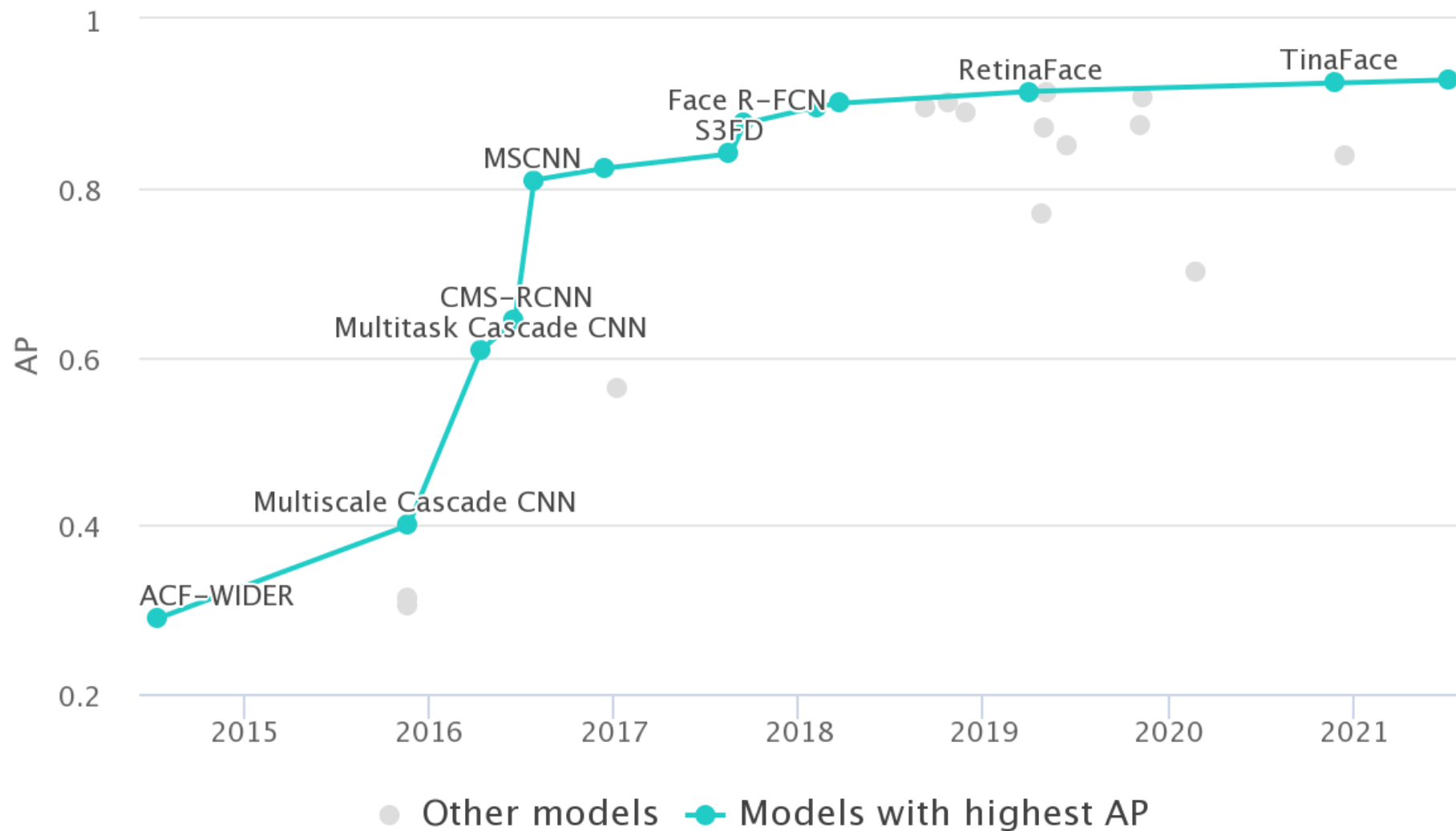
VJ [Viola 2004]
ACF [Dollar 2009]
DPM [Felz. 2010]
Faceness [Yang 2015]



Retinaface [Deng 2019]

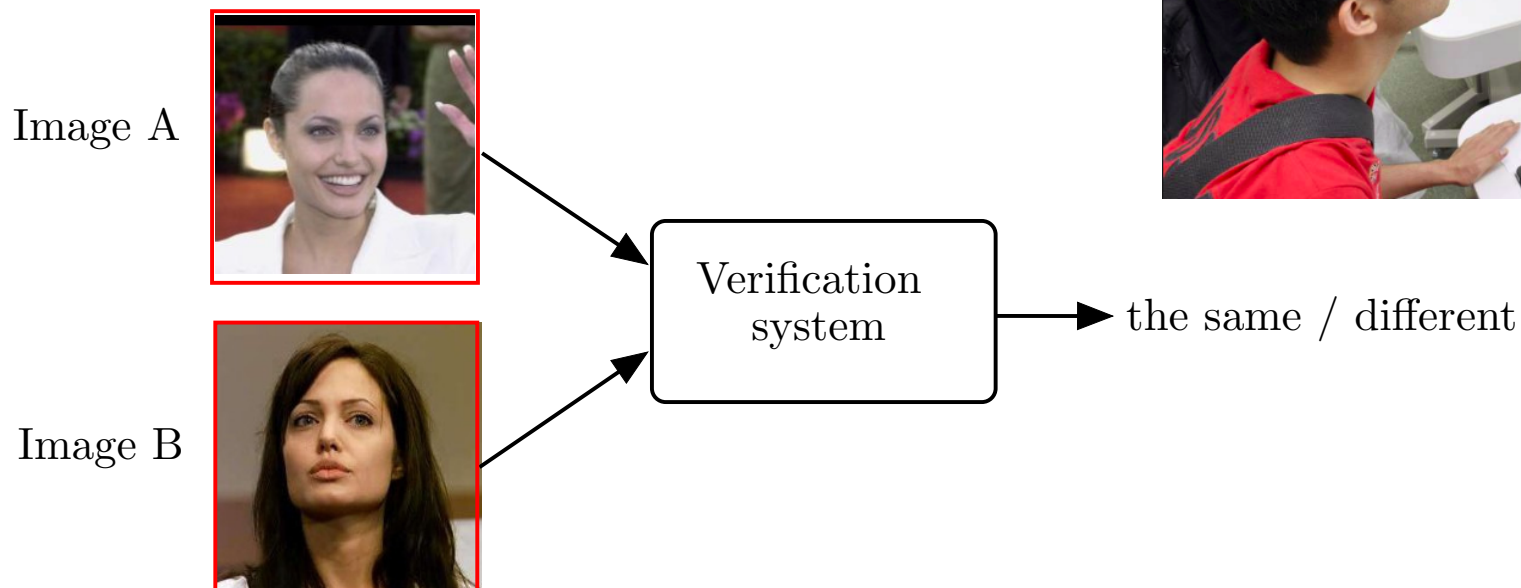
Přesnost detektorů na reálných datech

- ◆ Přesnost detektorů na “hard” části WIDER FACE databáze.



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”.
- ◆ **Aplikace:** přístup do kontrolovaných prostor (např. průchod letištěm viz obr), odemykání mobilních telefonů,...



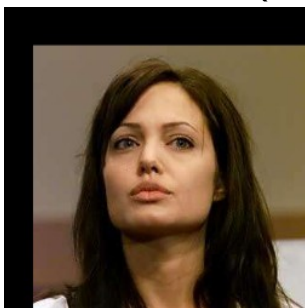
Měření přesnosti verifikačních metod

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

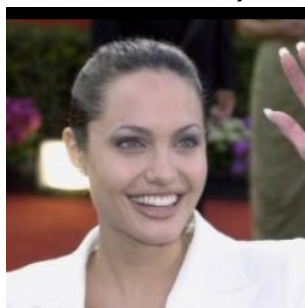
$$\{(A^1, B^1, y^1), \dots, (A^m, B^m, y^m)\}$$

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

$y = +1$ (stejná identita)

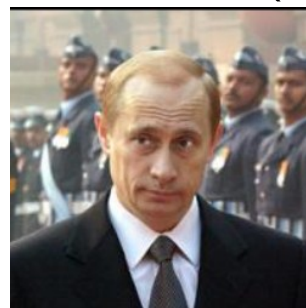


A

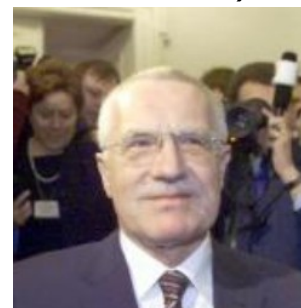


B

$y = -1$ (různé identity)



A



B

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

Metriky pro měření verifikačních algoritmů

◆ False Acceptance Rate, False Match Rate

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

Podíl pasažérů s falešným pasem, kteří projdou branou pasové kontroly.

◆ True Acceptance Rate, Recall

$$\text{TAR} = \frac{\text{počet pozitivních příkladů klasifikovaných jako pozitivní}}{\text{počet všech pozitivních příkladů}}$$

Podíl pasažérů s platným pasem, které brána pustí.

◆ Classification Accuracy – příklad špatné metriky pro tuto aplikaci

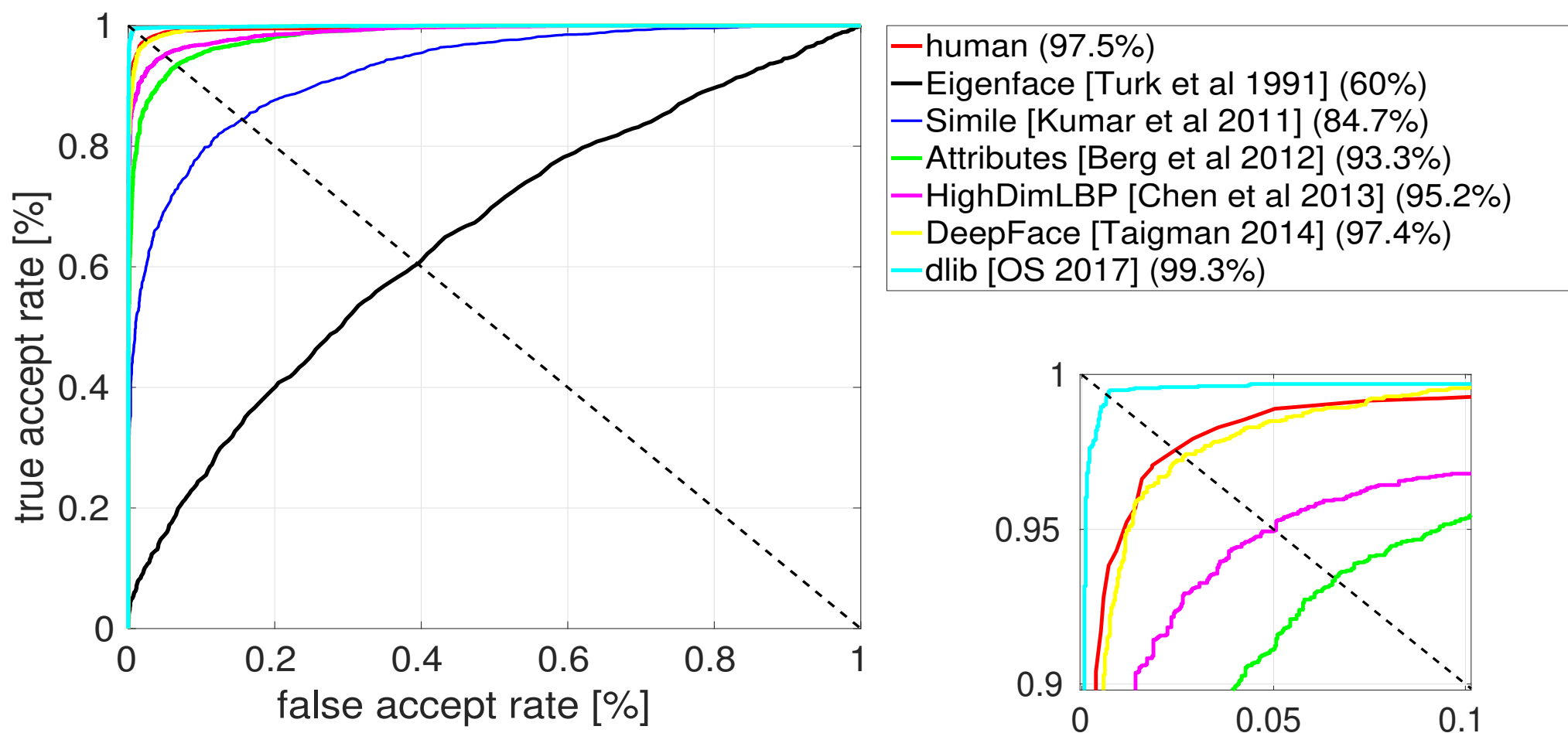
$$\text{ACC} = \frac{\text{počet chybně klasifikovaných příkladů}}{\text{počet všech příkladů}}$$

Přesnost verifikačních metod na LFW databázi

- Verifikační stroje rozhodují na základě prahování “dissimilarity” $s: I \times I \rightarrow \mathbb{R}$

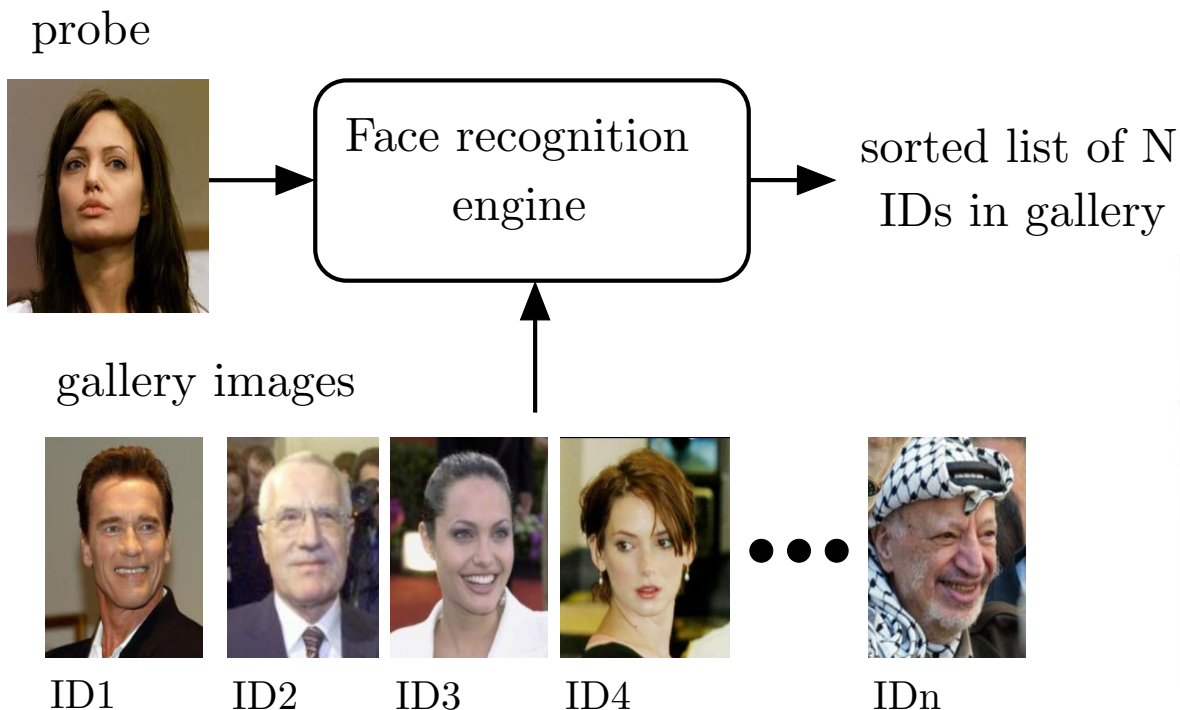
$$f_{\theta}(\text{obrázek A, obrázek B}) = \begin{cases} +1 & \text{pokud } s(\text{obrázek A, obrázek B}) \leq \theta \\ -1 & \text{pokud } s(\text{obrázek A, obrázek B}) > \theta \end{cases}$$

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



Úlohy rozpoznávání tváří: Vyhledávání tváře

- ♦ **Úloha:** vyhledat identitu v databázi na základě její tváře
- ♦ **Výstup:** seznam N identit seřazených podle míry podobnosti
 - “Close-set” scénář - hledaná identita je v databázi; N je fixní
 - “Open-set” scénář - hledaná identita nemusí být v databázi; N určen tak, aby obsahoval jen hledanou identitu
- ♦ **Aplikace:** zpracování záznamů z dohledových kamer, vyhledávání zajímavých osob v databázi, odstranění duplikátů (mugshot shot repos., ID card DBs)...



Eyedentity (www.eyedea.cz)



Metriky pro měření přesnosti vyhledávače tváří

◆ Rank-N (Close-set)

$$\text{RankN} = \frac{\text{kolikrát byla cílová identita mezi N prvními}}{\text{počet dotazů}}$$

◆ False Positive Identification Rate (Open-set)

$$\text{FPIR} = \frac{\text{počet vrácených seznamů s nenulovou délkou}}{\text{počet dotazů kdy cílová identita není v DB}}$$

Pravděpodobnost, že bude operátor muset zbytečně prověřovat výsledek.

◆ True Positive Identification Rate (Open-set)

$$\text{TPIR} = \frac{\text{počet dotazů kdy cílová identita je ve vráceném seznamu}}{\text{počet dotazů kdy cílová identita je v DB}}$$

Pravděpodobnost, že bude hledaná identity odhalena, pokud je v DB.

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



(a) Good quality mugshot



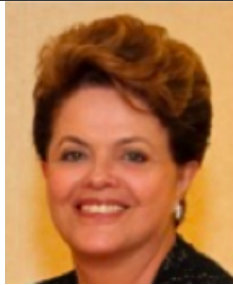
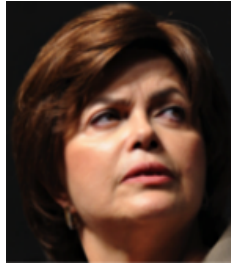
(b) Poor quality webcam

	Mugshots		WebCam
	Rank1 [%]	Rank50[%]	Rank1 [%]
NEC	95.9	97.4	88.7
Morpho	90.9	92.9	70.2
Toshiba	89.3	94.3	76.3
Cognitec	86.4	91.6	42.4

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

Přesnost verifikace a identifikace tváří na IJB-A databázi [Klare et al. 2015]

- ◆ Velká variabilita v orientaci tváře, geografické příslušnosti subjektů, osvětlení.
- ◆ 25,813 obrázků (5,399 statistických, 20,414 video frames), 500 identit.
- ◆ protokoly: verifikace, identifikace, detekce
- ◆ vstup jsou “templates” - sekvence tváří zobrazující jednu identitu

type of images	 Frontal, cooperative subj. controlled environment	 near frontal, uncoop. subj. min environment variation	 full variation in pose illumination, environment
automated detection	human performance	near human performance	cannot detect consistently
automated recognition	human performance	near human performance	cannot recognize

Method	Verification (TAR @ FAR)			Identification (TPIR@FPIR, RankN)			
	0.001	0.01	0.1	FPIR=0.01	FPIR=0.1	Rank1	Rank10
[Klare 2015]	0.20	0.41	0.63	0.047	0.24	0.44	-
[Parkhi 2015]	0.60	0.81	0.94	0.46	0.67	0.91	0.98
[Yang 2016]	0.88	0.94	0.98	0.82	0.92	0.96	0.97
[Cao 2017]	0.92	0.97	0.99	0.88	0.95	0.98	0.99
[Ranjan 2017]	0.94	0.97	0.98	0.92	0.96	0.97	0.99

IARPA Janus Benchmark-B (IJB-B) databáze [Whitelam et al. 2017]

- ◆ Velká variabilita v orientaci tváře, geografické příslušnosti subjektů, osvětlení.
- ◆ 76.8K obrázků (21.8K statických, 55K video frames), 1,845 identit.
- ◆ protokoly: verifikace, identifikace, detekce
- ◆ vstup jsou “templates” - sekvence tváří zobrazující jednu identitu

LFW
(2008)



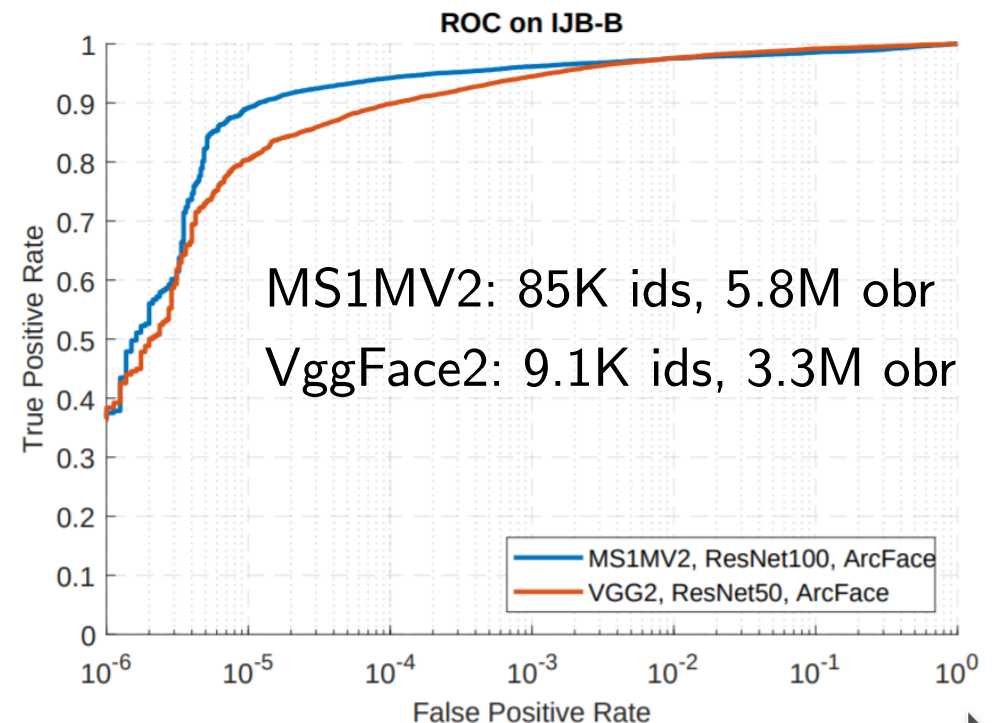
IJB-A
(2015)



IJB-B
(2017)



Method	Verifikace $\text{TAR@FAR}=10^{-4}$
VggFace2 [Cao 2018]	0.800
SENet50 [Xie 2018]	0.849
ArcFace [Deng 2019]	0.942



Úlohy rozpoznávání tváří: Odhadování věku

- ◆ Úloha: na základě obrázku tváře odhadnout 1) chronologický věk nebo 2) vizuální věk (apparent age)
- ◆ 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

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。



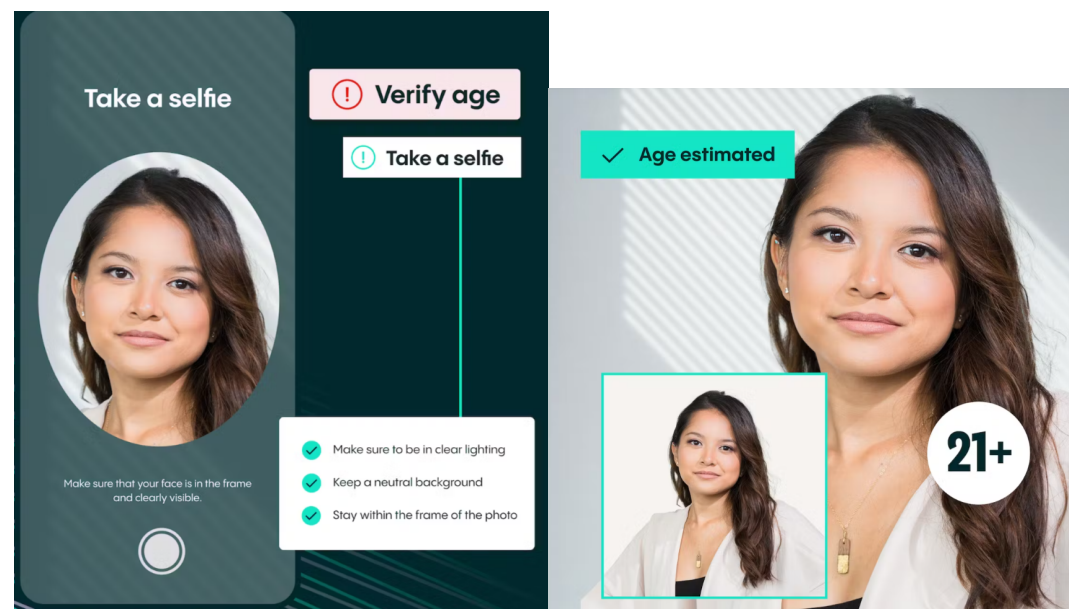
2.



3.

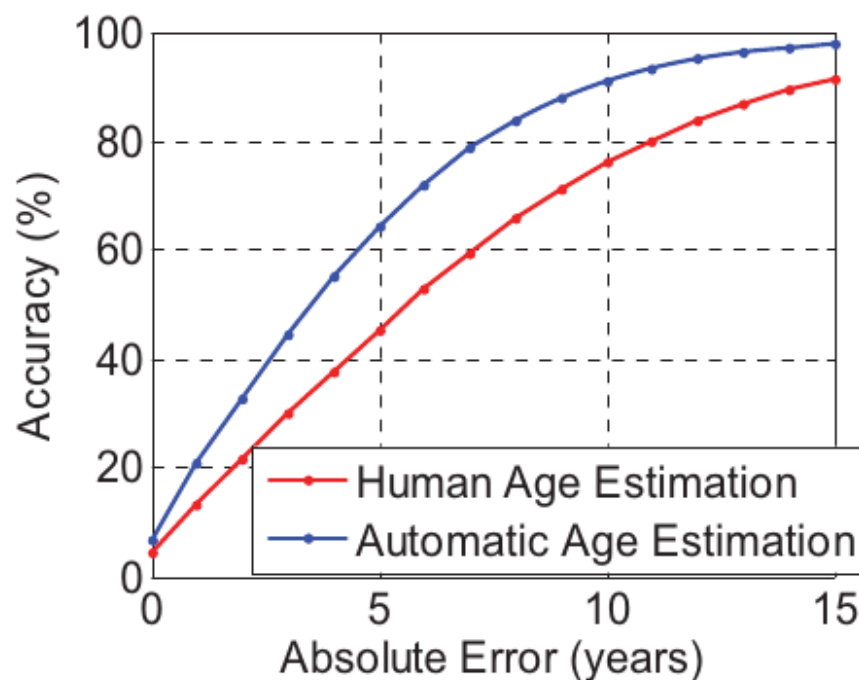


Vzdálená verifikace věku



Odhadování věku: stroj versus člověk

- ◆ Test na PSCO databázi 10,036 policejních fotografií s rovnoměrným zastoupením věku v rozmezí 17 až 68 let.
- ◆ Metriky: i) střední absolutní odchylka odhadnutého a skutečného věku (MAE), ii) procento tváří s absolutní odchylkou né horší než x let (CS_x score).
- ◆ Lidský odhad: 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.

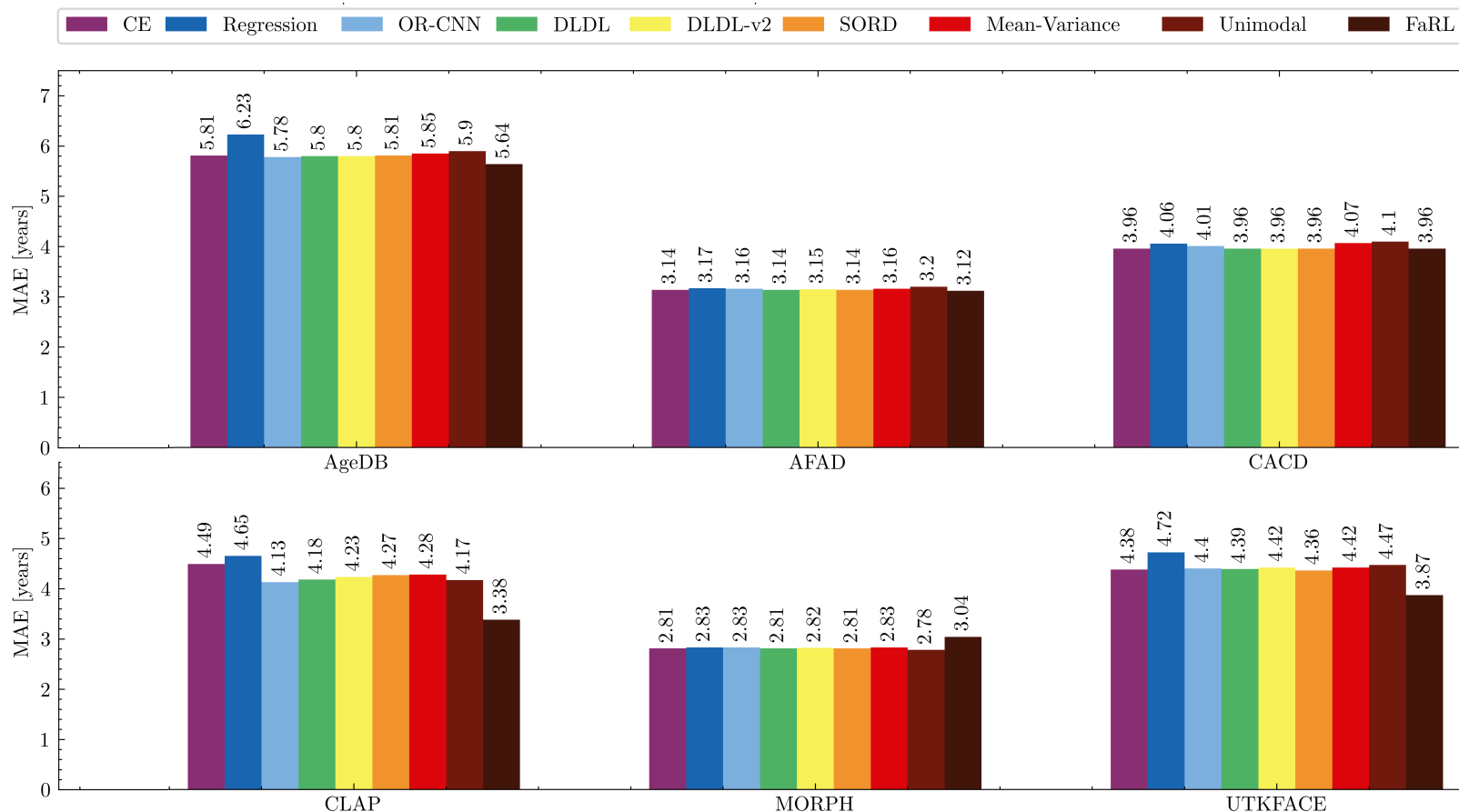


	MAE
Human	7.2
Machine	5.1

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

Odhadování věku: state-of-the-art 2024

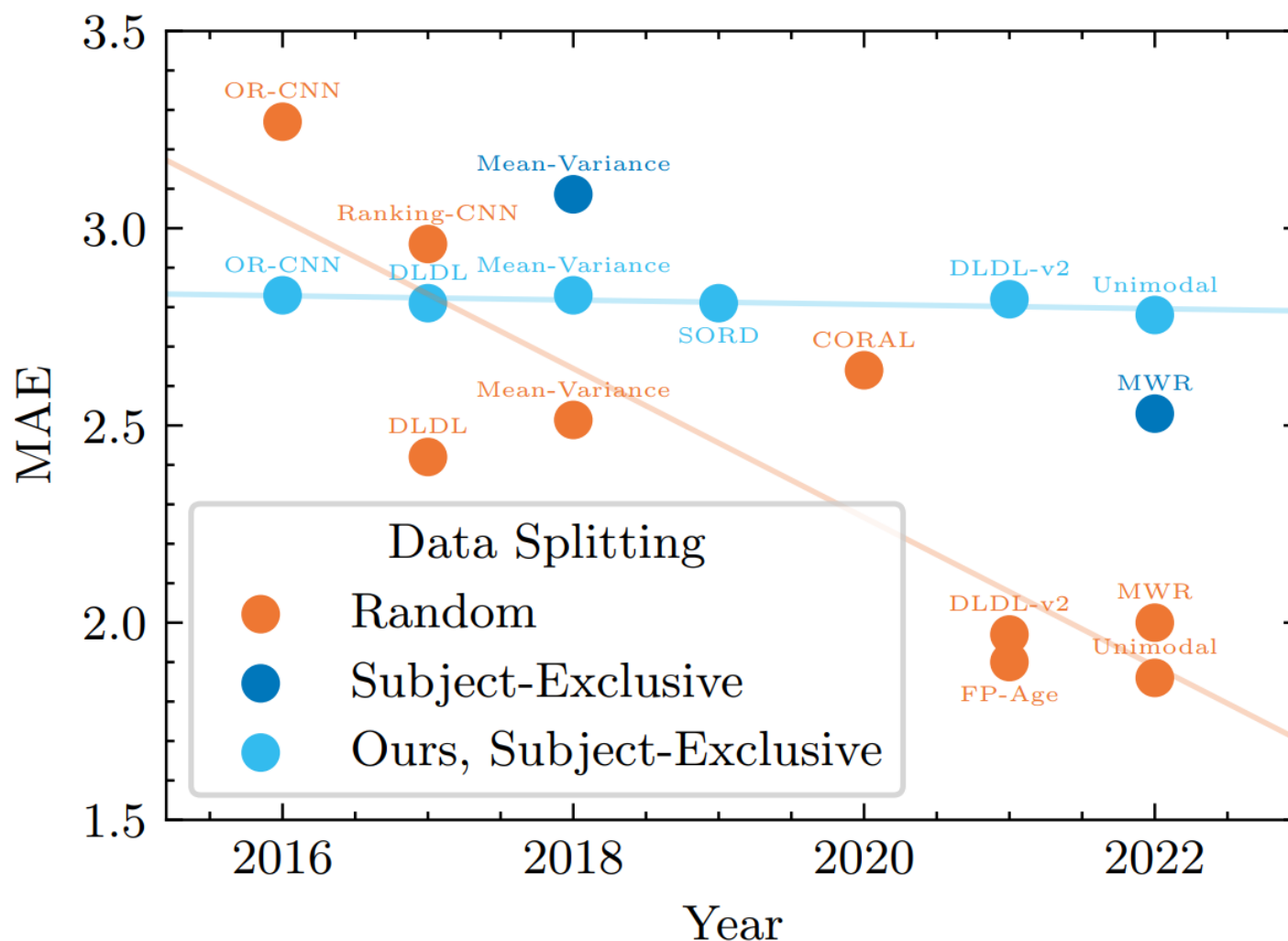
- ♦ Metoda: odhad věku postavený na konvolučních neuronových sítích.
- ♦ Databáze: AgeDB (16K), CACD (163K) - Celebrity images; MORPH (55k) - Police mugshots; AFAD (160K) - Asian faces from RenReb social network; CLAP (8K), UTKFace (20K) - In-the-wild photos from the Internet.
- ♦ Metrika: Střední absolutní odchylka (MAE) odhadnutého a skutečného věku.



Paplá, Franc: Unraveling the Age Estimation Puzzle: Comparative Analysis of Deep Learning Approaches for Facial Age Estimation. CVPR. 2024.

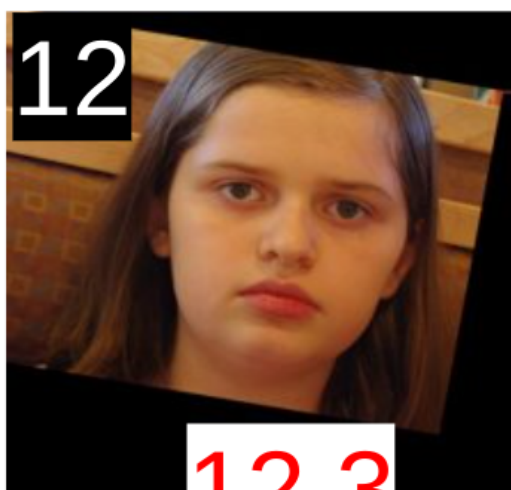
Odhadování věku: důležitost správného testovacího protokolu

- ◆ MORPH databáze policejních fotografií; 55K fotografií, 13K identit.
- ◆ Neexistuje standardní protokol rozdělování data na trénovací a testovací. Různí autoři používají buď: i) náhodné rozdělení obrázků nebo ii) náhodné rozdělení identit.

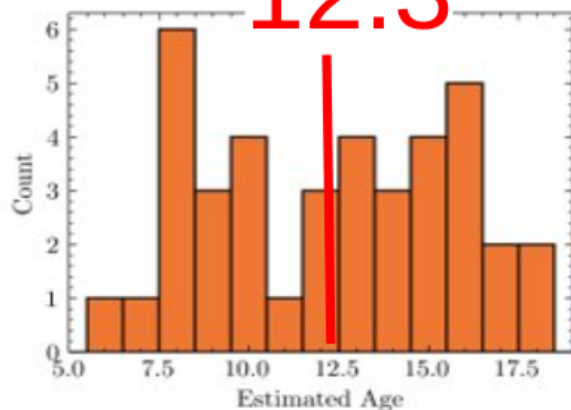


Odhadování věku: rozdíl mezi chronologickým a vizuálním věkem

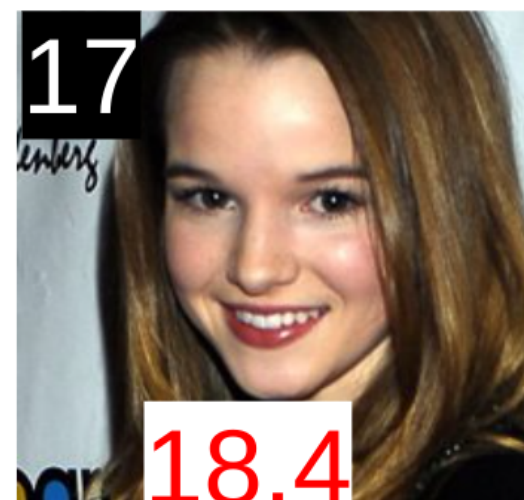
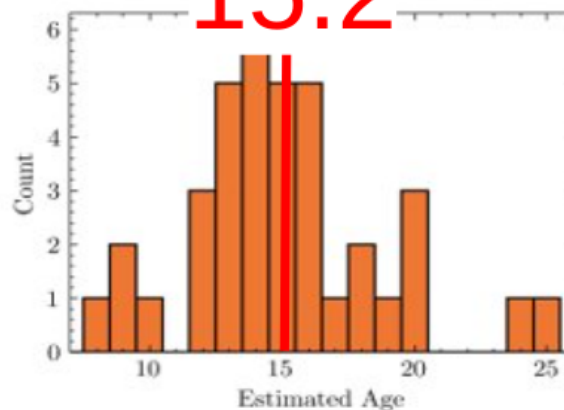
- **Task 1 (main):** Real age estimation
- **Task 2:** Apparent age estimation



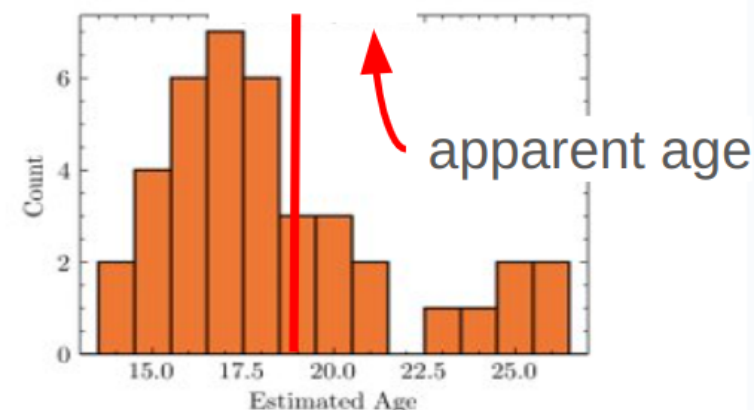
12.3



15.2



18.4

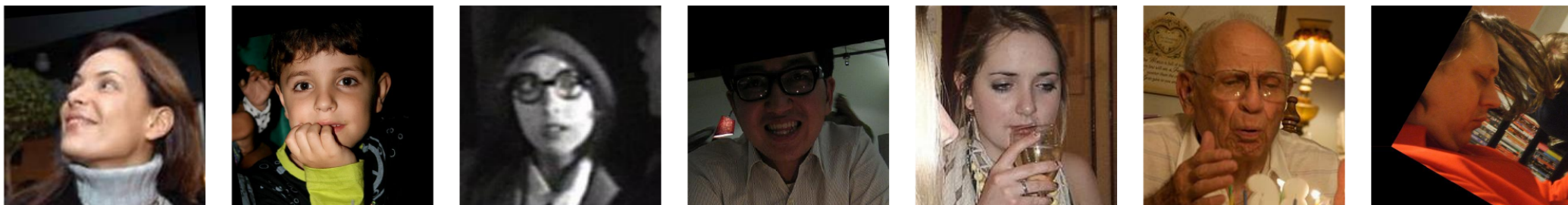


real age

apparent age

Odhadování věku: člověk vs. stroj 2024

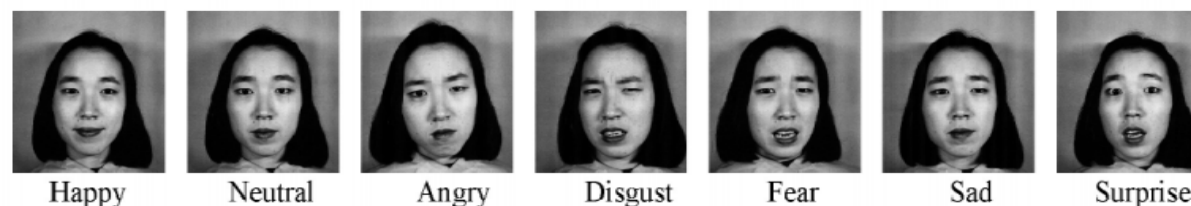
- ◆ APPA-REAL databáze 7,591 “in-the-wild” obrázků lidských tváří.
- ◆ Každá tvář anotovaná chronologickým věkem (0-100) a cca 30+ lidskými odhady věku (celkem 260K lidských odhadů).
- ◆ Metrika: Střední absolutní odchylka (MAE) odhadnutého a skutečného věku.



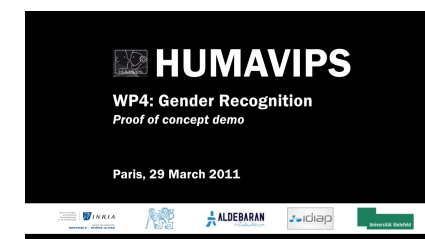
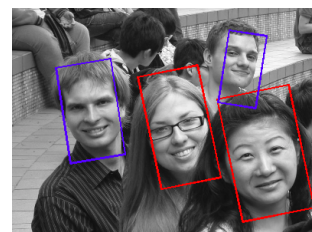
	Apparent Age (= avg of 30+ human age estimates)		Real Age	
	[0 – 18] Years	[0 – 90] Years	[0 – 18] Years	[0 – 90] Years
CTU-FaceAge	1.8	3.6	2.7	4.9
Human	1.4	3.9	3.2	6.0
Previous SOTA	NA	4.1	NA	5.2

Některé další úlohy rozpoznávání tváří

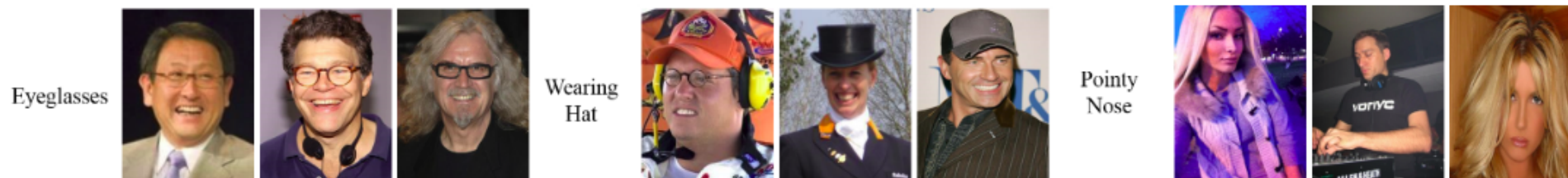
- ◆ **Rozpoznávání emocí.** Výstup: happy, sad, fear, disgust, surprise, anger, neutral.
Úloha: klasifikace. Metrika: klasifikační chyba.



- ◆ **Rozpoznání pohlaví.** Výstup: muž, žena. Úloha: klasifikace. Metrika: klasifikační chyba.

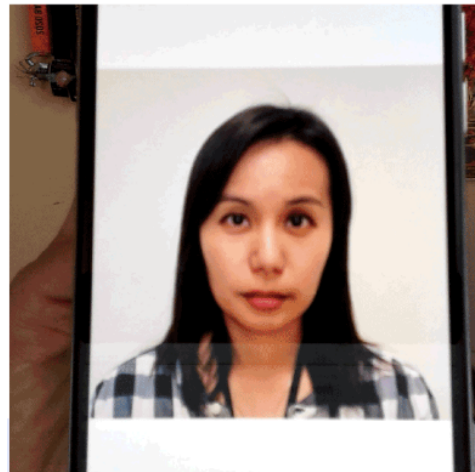


- ◆ **Face attribute recognition:** eye-glasses, wearing-hat, pointy-nose, mustache, ...
Úloha: multi-label classification. Metrika: klasifikační chyba pro každou třídu zvlášť.



Úlohy rozpoznávání tváří: Face anti-spoofing

- ◆ Cílem je zjistit zdali před kamerou stojí živá tvář.
- ◆ Typy útoků: “print attack”, “replay attack”, “3D mask attack”, “make-up attack”, ...
- ◆ Vstup: Video tváře.
- ◆ Výstup: live, spoofed.
- ◆ Metrika: FPR, TPR.



Úlohy rozpoznávání tváří: Hodnocení kvality tváře

- ◆ Cílem je odhadnout zdali systém pro rozpoznávání (např. verifikátor) na dané tváři selže, předtím než systém použijeme.
- ◆ Skóre kvality tváře: skálární veličina korelující s pravděpodobností selhání rozpoznávacího systému.
- ◆ Benchmark: NIST organizuje FRVT – Quality Summarization and Analysis challenge.
- ◆ Aplikace: zajištění vysoké přesnosti, výběr reprezentativní tváře, atd.



Q = 95

Q = 85

Q = 62

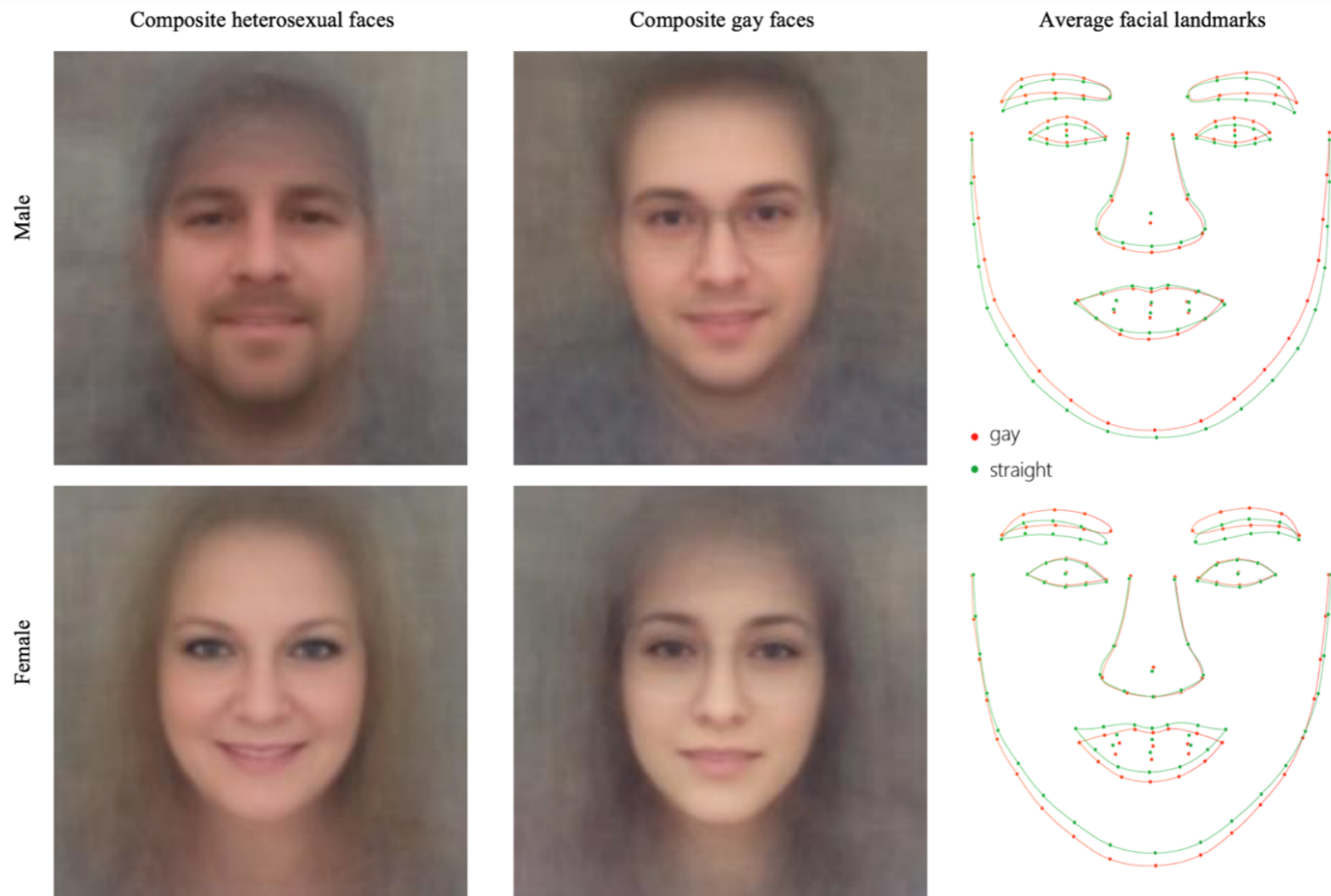
Q = 42

https://pages.nist.gov/frvt/html/frvt_quality.html

Detekce sexuální orientace z tváře

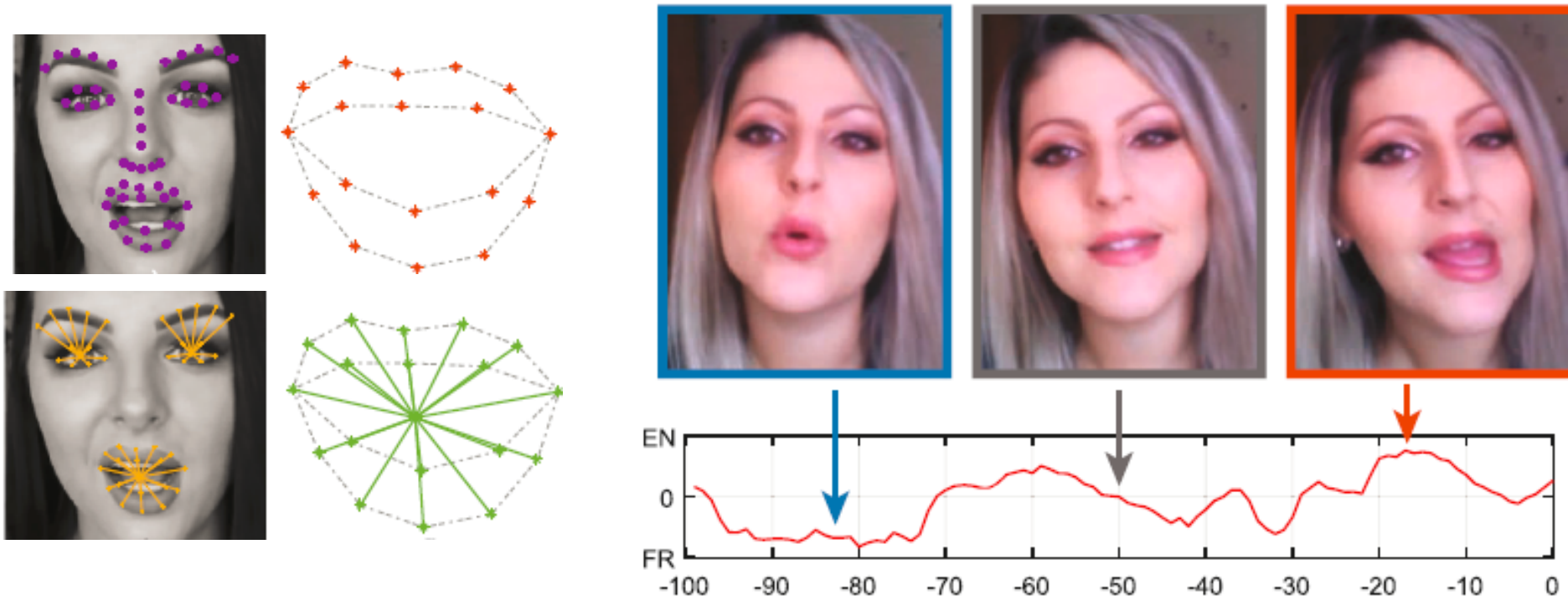
[M.W.Kosinski et al. 2018]

- ◆ 35,326 tváří 14,776 gay a heterosexuálních mužů (50%/50%) a žen (53%/47%) stažených z online seznamky.
- ◆ CNN+logistic regression: přesnost klasifikace 81% u mužů 71% u žen.
- ◆ Lidský odhad: přesnost 61% u mužů a 54% žen.



Úlohy rozpoznávání tváří z videa: Rozpoznávání jazyka

- ◆ Úloha: odhadnout jazyk mluvčího pouze z videa
- ◆ Třídy: angličtina, francouzština

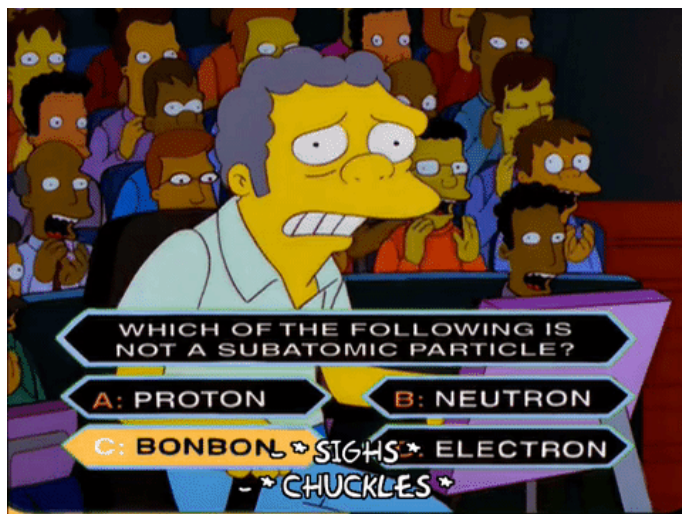


- ◆ Klasifikační chyba stroje je 27%. Chyba člověka je 28%.

Špetlík, Čech, Franc, Matas: *Visual Language Identification from Facial Landmarks*. SCIA, 2017.

Úlohy rozpoznávání tváří z videa: Detekce nejistoty

- ♦ Úloha: odhadnout nejistotu při rozhodování na základě videa tváře
- ♦ Data z populární soutěže “Who wants to be a millionaier?”
- ♦ Třídy: nejistota = soutěžící požádá o nápovědu, jistota = soutěžící odpoví bez nápovědy



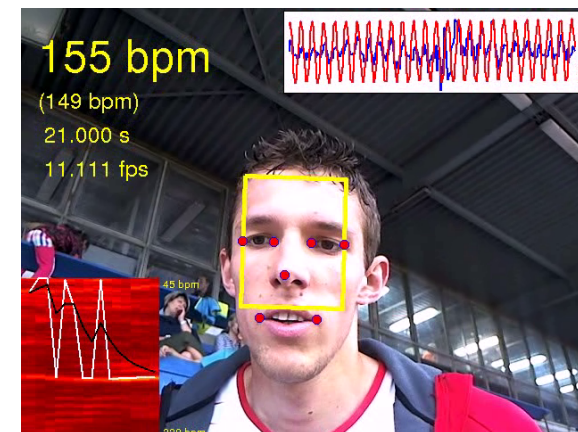
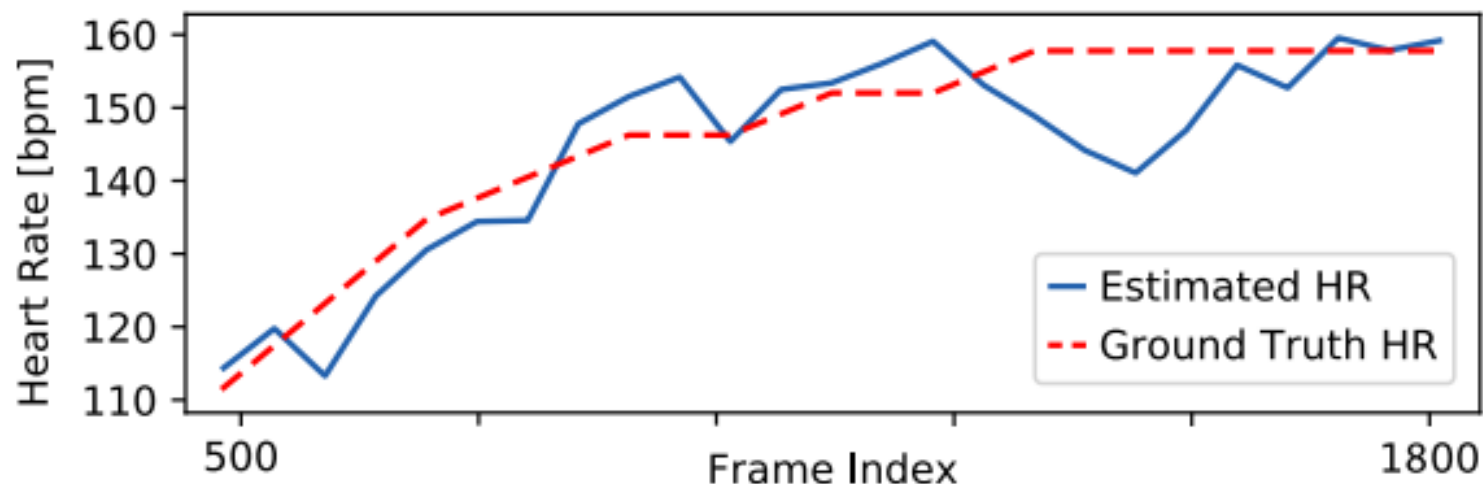
- ♦ Klasifikační chyba stroje je 25%. Chyba člověka je 45%.

Jahoda, Vobecký, Čech, Matas: Detection decision ambiguity from facial images. IEEE face and Gesture, 2018.

Úlohy rozpoznávání tváří z videa: Odhad tepové frekvence

- ♦ Úloha: odhadnout tepovou frekvenci na základě videa sledovaného subjektu.
- ♦ Výstup: tepová frekvence subjektu.
- ♦ Metrika: Mean Absolut Error (MAE)

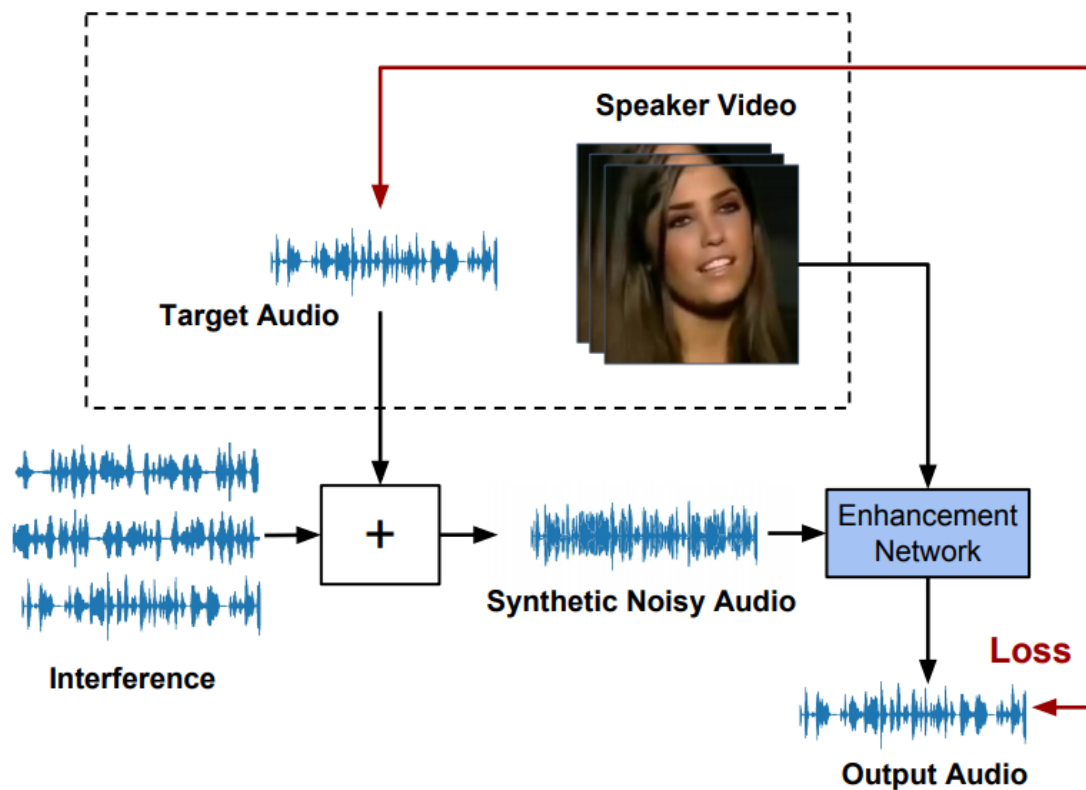
		COHFACE	ECG-Fitness	MAHNOB	PURE	PURE MPEG-4 Visual
MAE [bpm]	2SR	20.98	43.66	13.84	2.44	5.78 ①
	CHROM	7.80 ①	21.37	13.49	2.07	6.29
	LiCVPR	19.98	63.25	7.41	28.22	28.39
	SAMC*	—	—	—	—	—
	HR-CNN	8.10	14.48 ①	7.25 ①	1.84 ①	8.72



Spetlik, Franc, Cech, Matas: Visual Heart Rate Estimation with Convolutional Neural Network. BMVC, 2018.

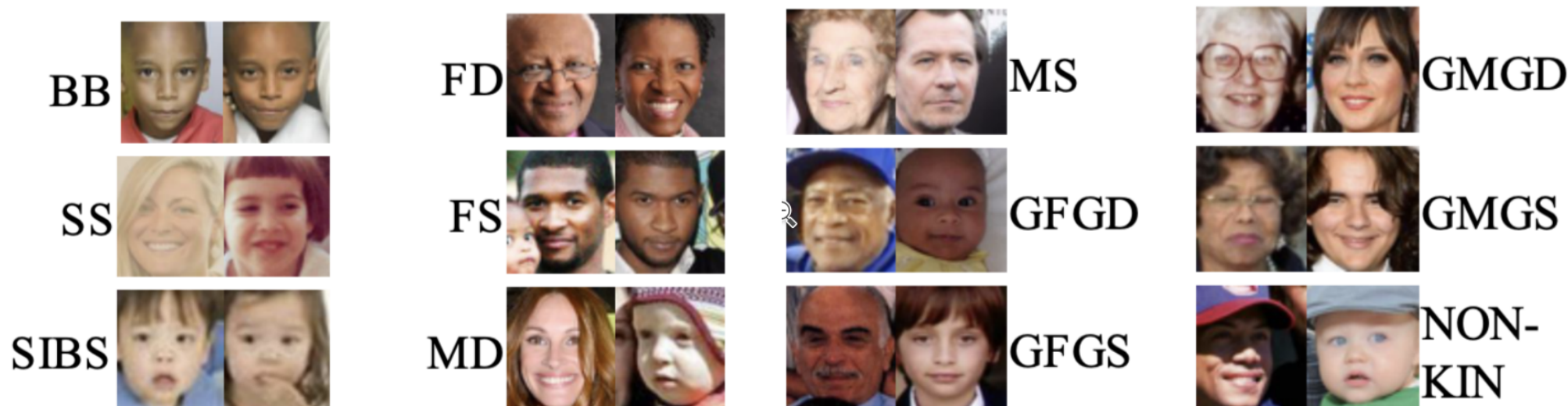
Úlohy rozpoznávání tváří z video a audia: Speech enhancement

- ♦ Úloha: zvýraznit audio subjektu vybraného podle jeho tváře
- ♦ Výstup: audio



Afouras, Chung and Zisserman: The Conversation: Deep Audio-Visual Speech Enhancement. Interspeech, 2018.

Úlohy rozpoznávání tváří: Kinship verification

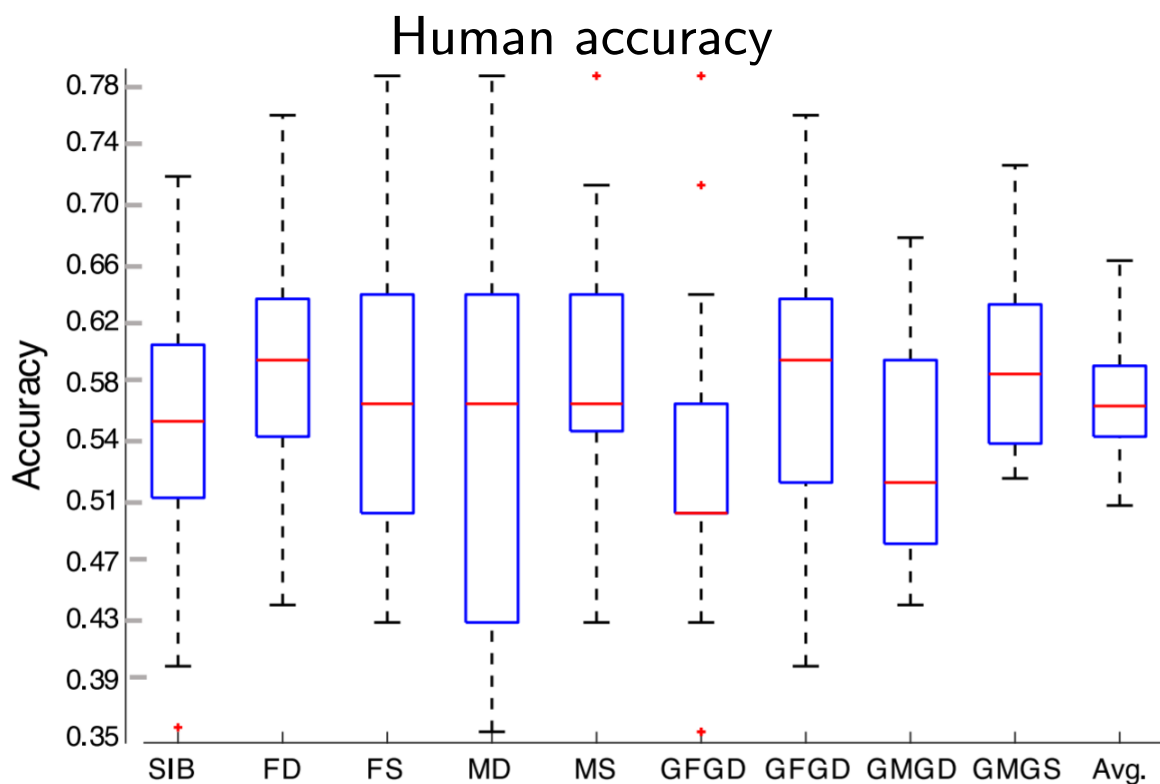
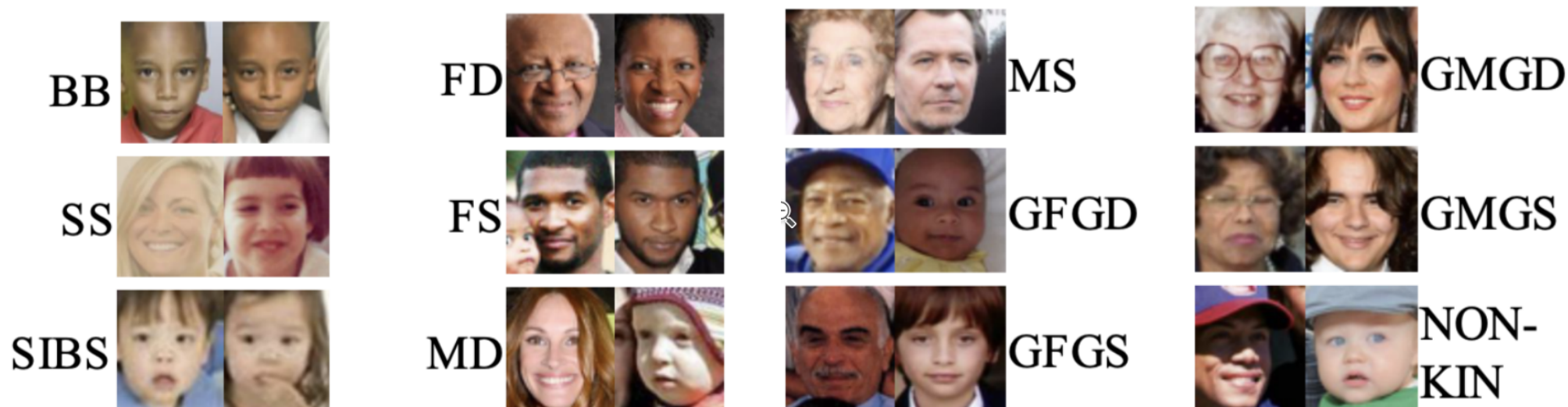


Averaged Verification Accuracies (%) for 5-Fold Experiment on FIW with No Family Overlap Between Folds

Method	siblings			parent-child				grandparent-grandchild				Acc. $\pm$ Std.
	B-B	S-S	SIBS	F-D	F-S	M-D	M-S	GF-GD	GF-GS	GM-GD	GM-GS	
LBP [56]	55.52	57.49	55.39	55.05	53.77	55.69	54.65	55.79	55.92	54.00	55.36	55.33 $\pm$ 1.01
SIFT [57]	57.86	59.34	56.91	56.37	56.24	55.05	56.45	57.25	55.35	57.29	56.74	56.80 $\pm$ 1.17
ResNet-22 [58]	65.57	69.65	60.12	59.45	60.27	61.45	59.37	55.37	58.15	59.74	59.70	61.34 $\pm$ 3.81
VGG-Face [36]	69.67	75.35	66.52	64.25	63.85	66.43	62.80	62.06	63.79	57.40	61.64	64.89 $\pm$ 4.68
+ITML [59]	57.15	61.61	56.98	58.07	54.73	57.26	59.09	62.52	59.60	62.08	59.92	59.00 $\pm$ 2.44
+LPP [60]	67.61	66.22	71.01	62.54	61.39	65.04	63.54	63.50	59.96	60.00	63.53	64.03 $\pm$ 3.32
+LMNN [61]	67.11	68.33	66.88	65.66	67.08	68.07	66.16	61.90	60.44	63.68	60.15	65.04 $\pm$ 3.00
+GmDAE [62]	68.05	68.55	67.33	66.53	68.30	68.15	66.71	62.10	63.93	63.84	63.10	66.05 $\pm$ 2.36
+DLML [63]	68.03	68.87	67.97	65.96	68.00	68.51	67.21	62.90	63.96	63.11	63.55	66.19 $\pm$ 2.36
+mDML [41]	69.10	70.15	68.11	67.90	66.24	70.39	67.40	65.20	66.78	63.11	63.45	67.07 $\pm$ 2.44
ResNet+CF [64]	69.88	69.54	69.54	68.15	67.73	71.09	68.63	66.37	66.45	64.81	64.39	67.87 $\pm$ 2.15
SphereFace [65]	71.94	77.30	70.23	69.25	68.50	71.81	69.49	66.07	66.36	64.58	65.40	69.18 $\pm$ 3.68

Robinson et. al.: Visual Kinship Recognition of Families in the wild. PAMI 2018.

Úlohy rozpoznávání tváří: Kinship verification



Robinson et. al.: Visual Kinship Recognition of Families in the wild. PAMI 2018.

Rozpoznávání tváří: porovnání s ostatními biometrickými metodami

Silné stránky:

- ◆ Nevyžaduje kooperující subjekt na rozdíl 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 spolehlivé v porovnání s technologiemi jako rozpoznávání otisků nebo duhovky.
- ◆ Dosažení vysoké efektivity není jednoduché: buď je třeba zajistit kontrolované podmínky nebo specializovaný hardware nebo velké množství dat.
- ◆ Robustní detekce “spoof attack” je stále otevřený problém.
- ◆ Robustní detekce selhání systému rozpoznávání tváří je stále otevřený problém.

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, měřítka, rotace
 - změnou osvětlení
 - změnou výrazu tváře (úsměv, smutek, neutrální, ...)
 - zákryty (brýle, pokrývka hlavy)
 - změna účesu, make up, stárnutím ...
 -

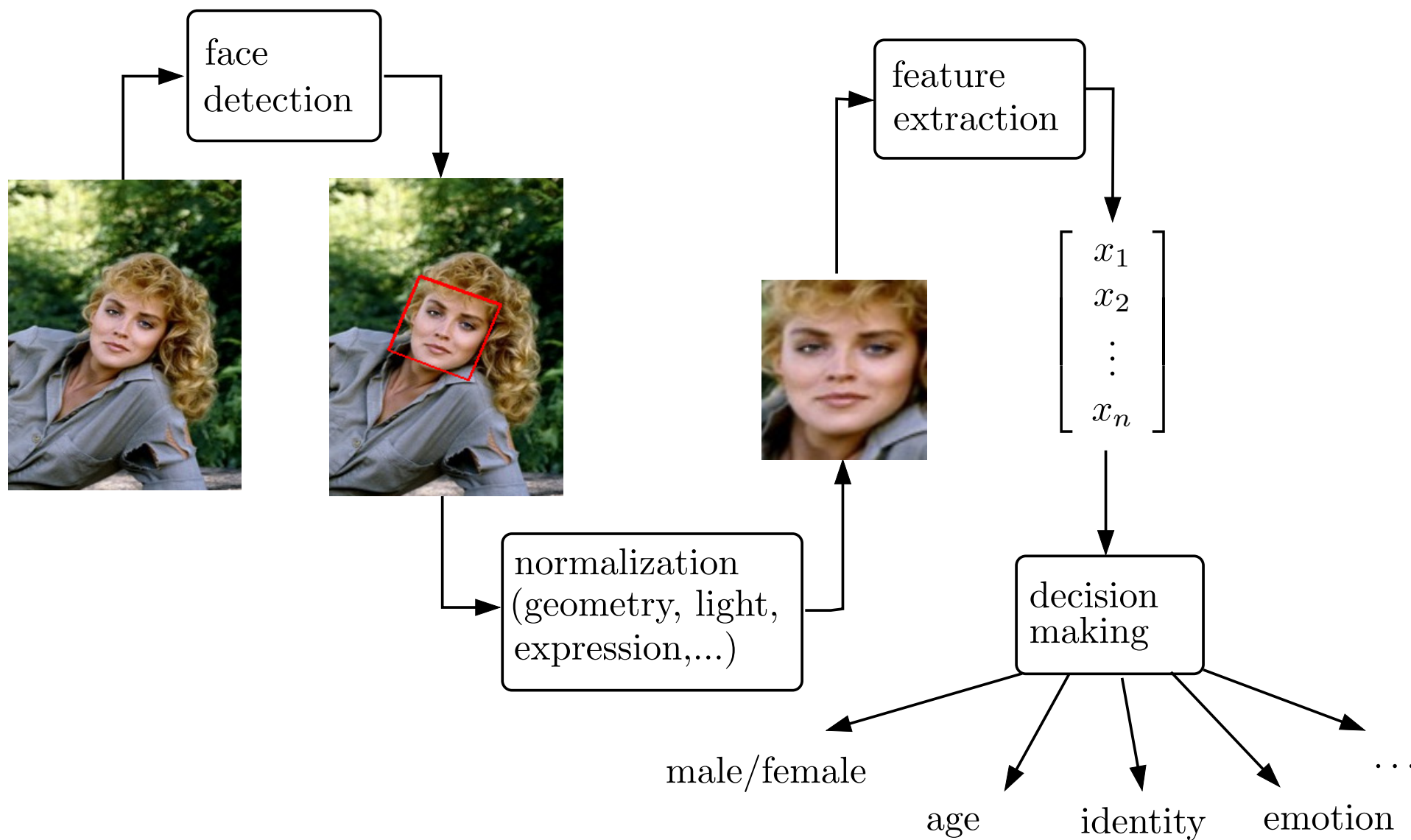


Funkčnost systému rozpoznávání tváří je ovlivněna mnoha faktory

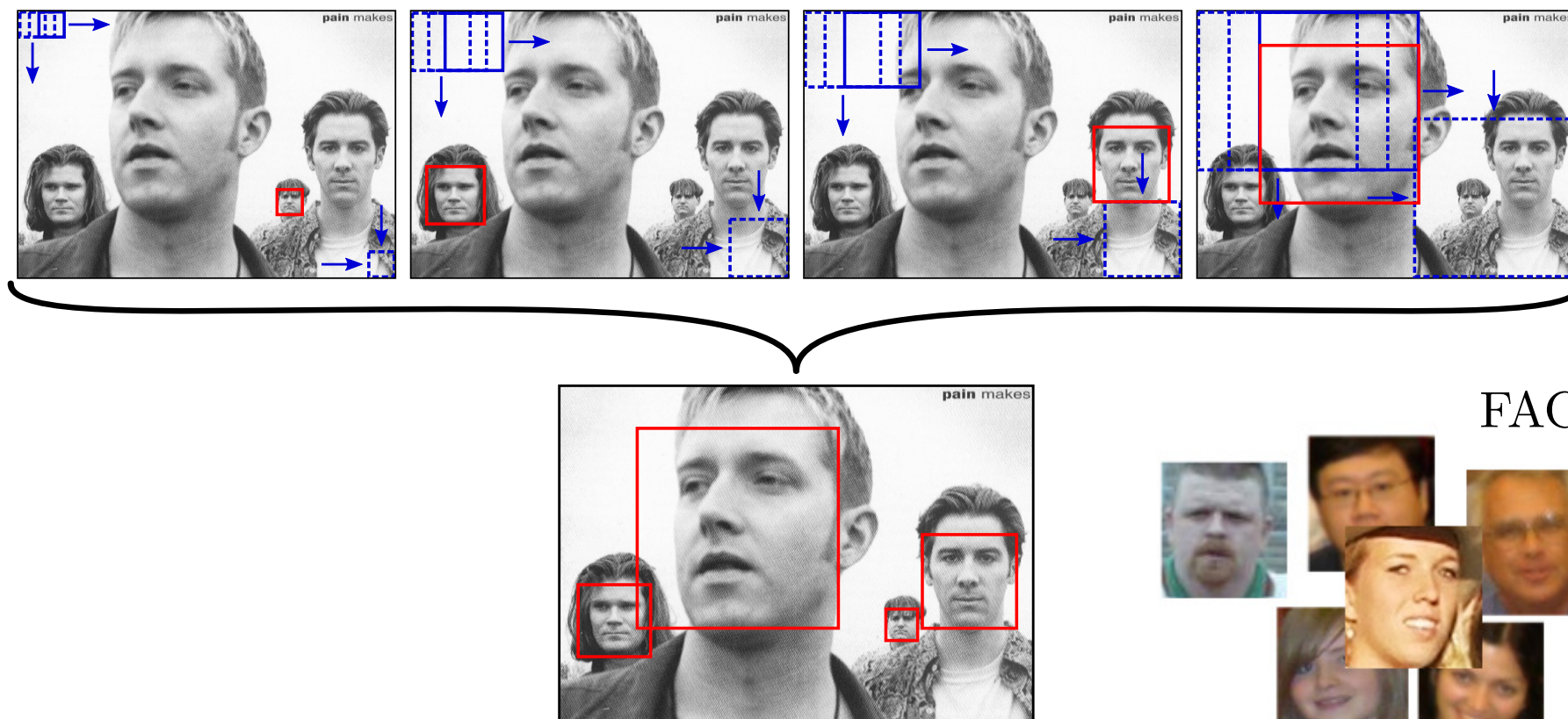
- ◆ Použitý snímací senzor:
 - kamera pracující ve viditelném spektru
 - infračervená kamera (s IR přísvícením)
 - stereo kamera
 - 3D skener
- ◆ Světelné podmínky (venkovní/vnitřní prostor, stíny, ...)
- ◆ Rozlišení obrazu a použitá komprese
- ◆ Statický obrázek / video sekvence
- ◆ Spolupracující /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



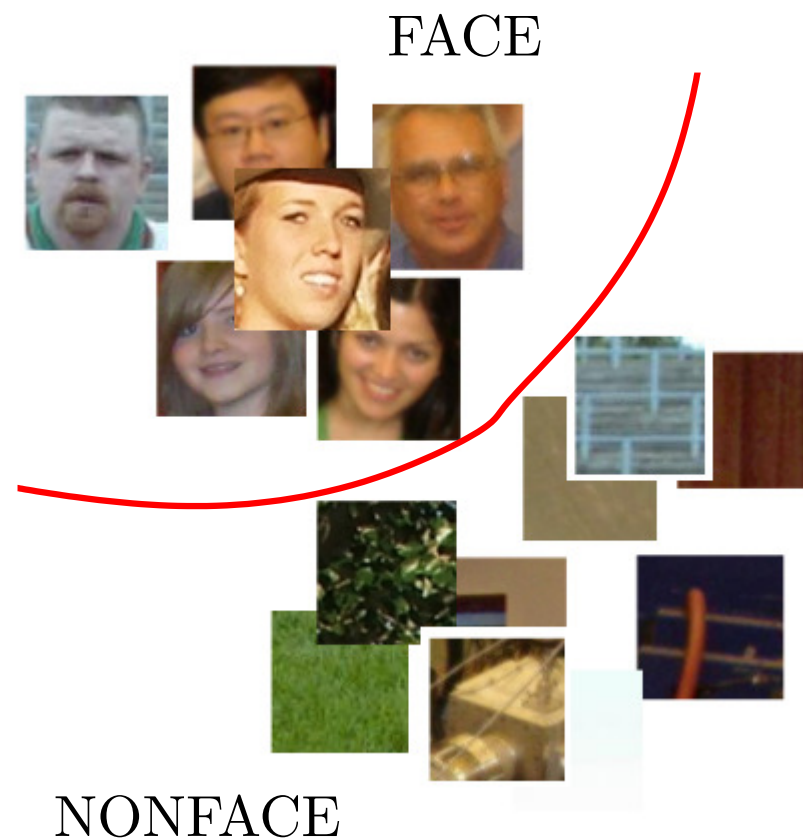
Stavební bloky systému pro rozpoznávání tváří



Detektor tváří



Problém detekce se převede na mnoho binárních klasifikací H : Obrázek $\rightarrow$ {tvář, netvář}

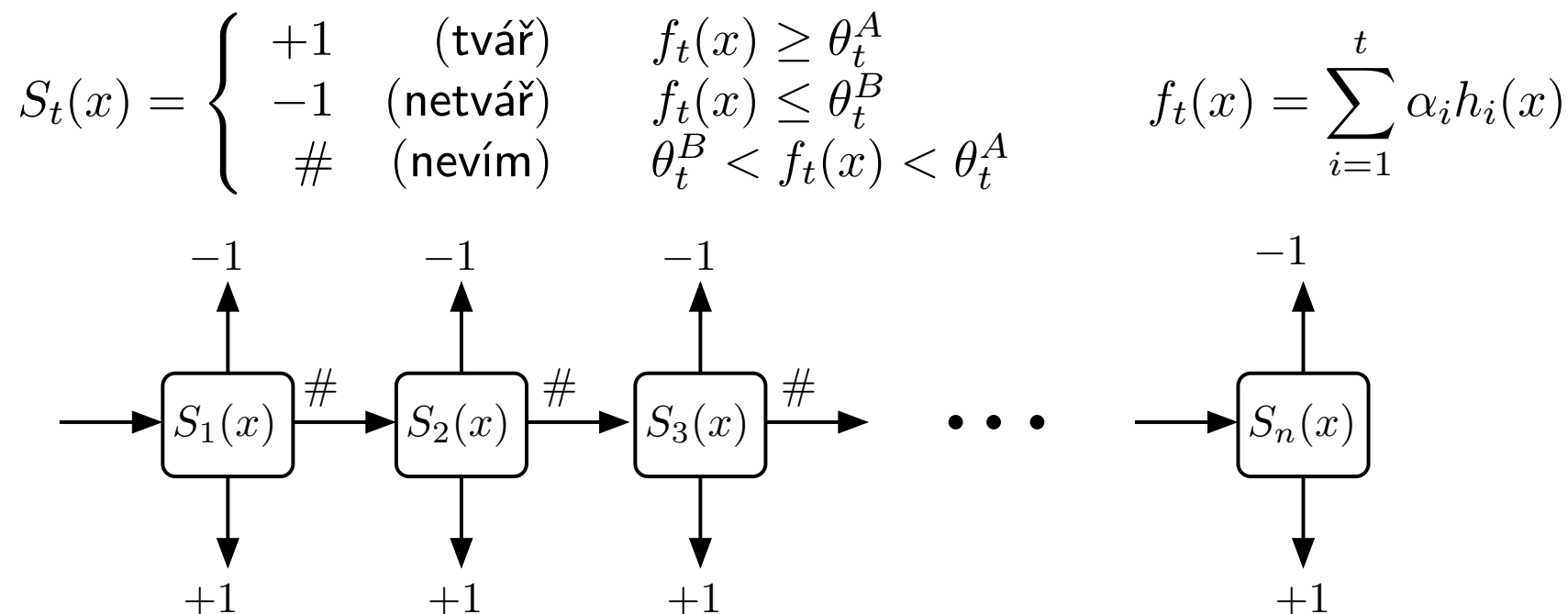


Klasifikátor tvář/netvář pro detekci tváří

- ◆ **Požadavky na klasifikátor:** extrémně přesný a extrémně rychlý.
 - Např. nalezení tváře velikosti 24×24 pixelů v obrázku 640×480 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: FP $\rightarrow$ 0% (chybné detekce), TP $\rightarrow$ 100% (detekované tváře)

Klasifikátor tvář/netvář pro detekci tváří

- ◆ **Požadavky na klasifikátor:** extrémně přesný a extrémně rychlý.
 - Např. nalezení tváře velikosti 24×24 pixelů v obrázku 640×480 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: FP $\rightarrow 0\%$ (chybné detekce), TP $\rightarrow 100\%$ (detekované tváře)
- ◆ **Řešení:** sekvenční rozhodovací pravidlo složené z jednoduchých rychlých klasifikátorů



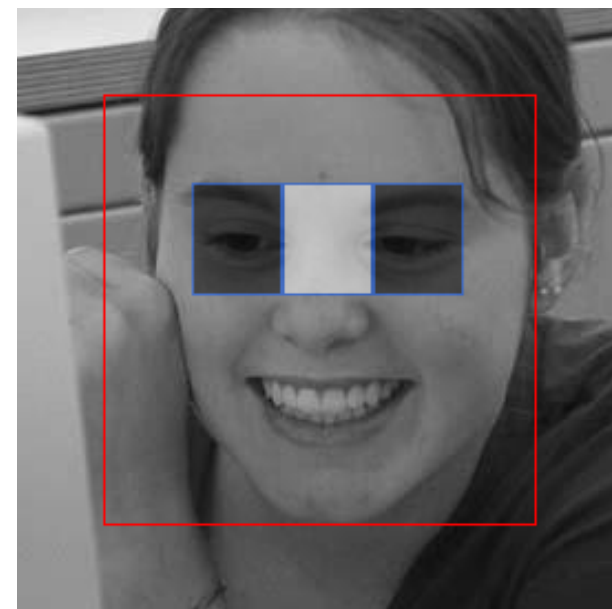
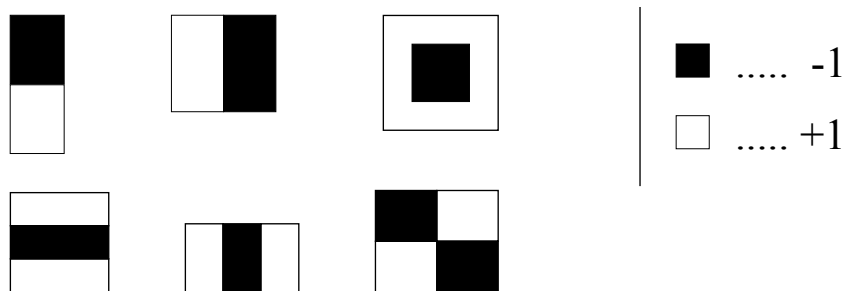
AdaBoost: sestroj silný klasifikátor ze slabých

Silný (přesný) klasifikátor

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

je složen ze slabých, ale rychlých klasifikátorů

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



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



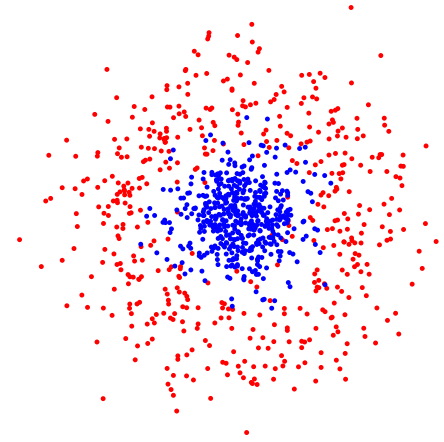
$(x_i, y_i = +1)$



$(x_i, y_i = -1)$

AdaBoost algoritmus

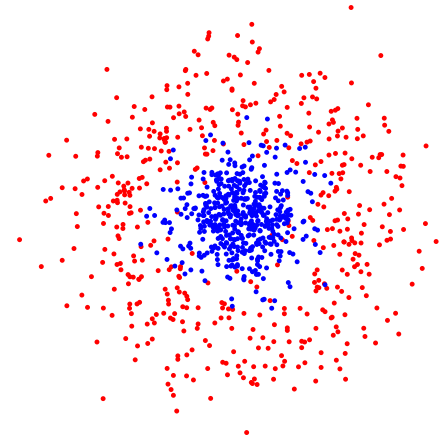
Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$



AdaBoost algoritmus

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

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

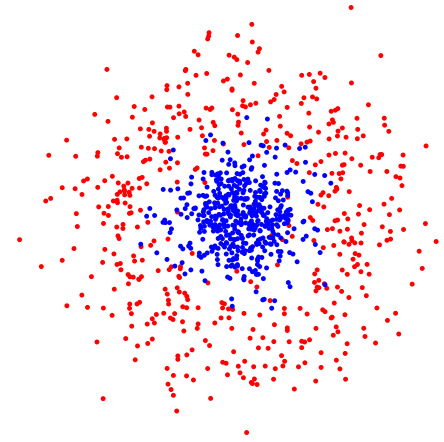


AdaBoost algoritmus

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

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

For $t = 1, \dots, T$:



AdaBoost algorithmus

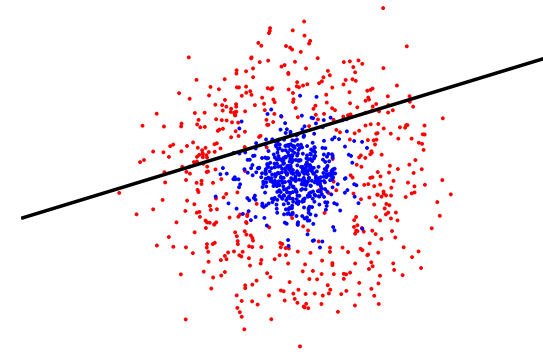
Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

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

For $t = 1, \dots, T$:

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

$t = 1$



AdaBoost algorithmus

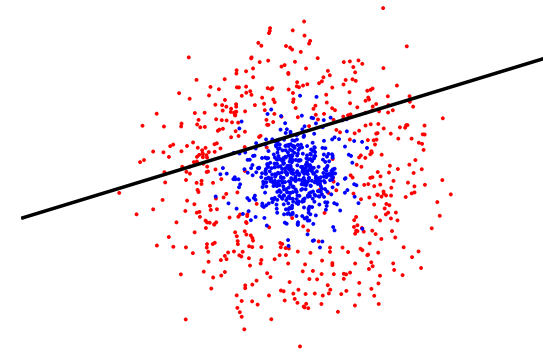
Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

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

For $t = 1, \dots, T$:

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

$t = 1$



AdaBoost algorithmus

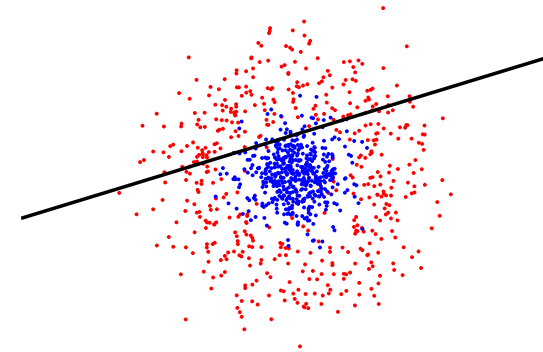
Given: $(x_1, y_1), \dots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$

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

For $t = 1, \dots, T$:

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

$t = 1$



AdaBoost algorithmus

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

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

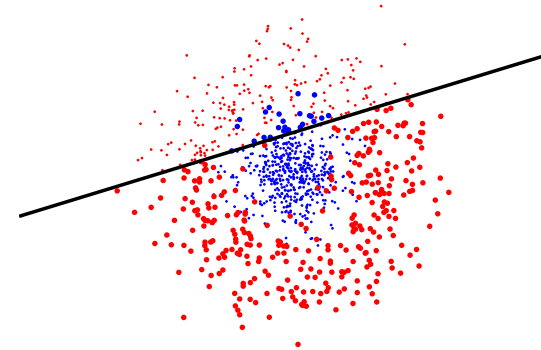
For $t = 1, \dots, T$:

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

$t = 1$



AdaBoost algorithmus

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

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

For $t = 1, \dots, T$:

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

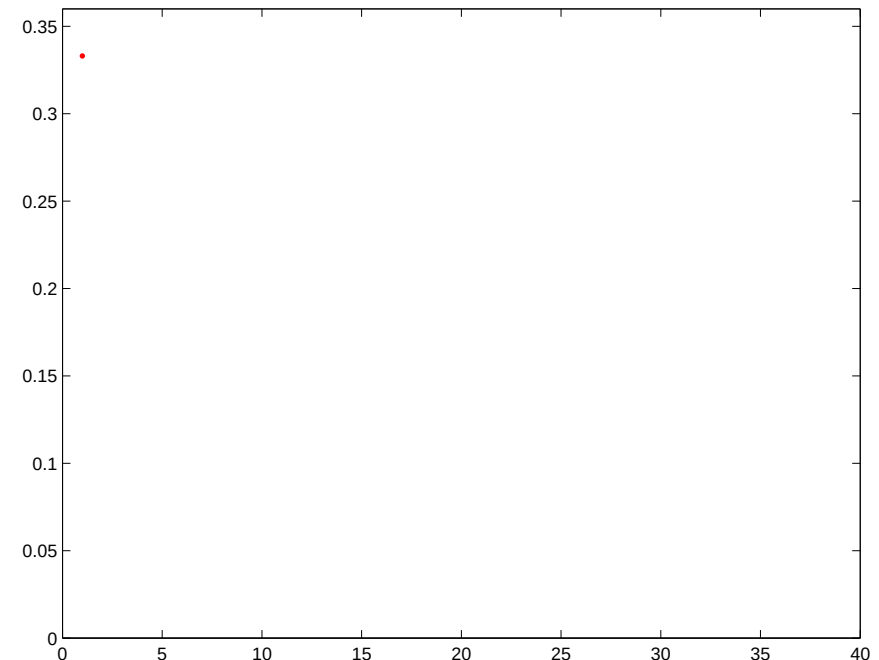
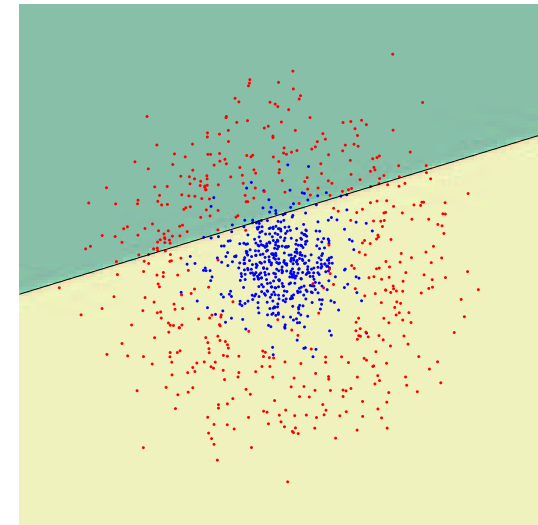
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is normalisation factor

Output the final classifier:

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

$t = 1$



AdaBoost algorithmus

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

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

For $t = 1, \dots, T$:

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

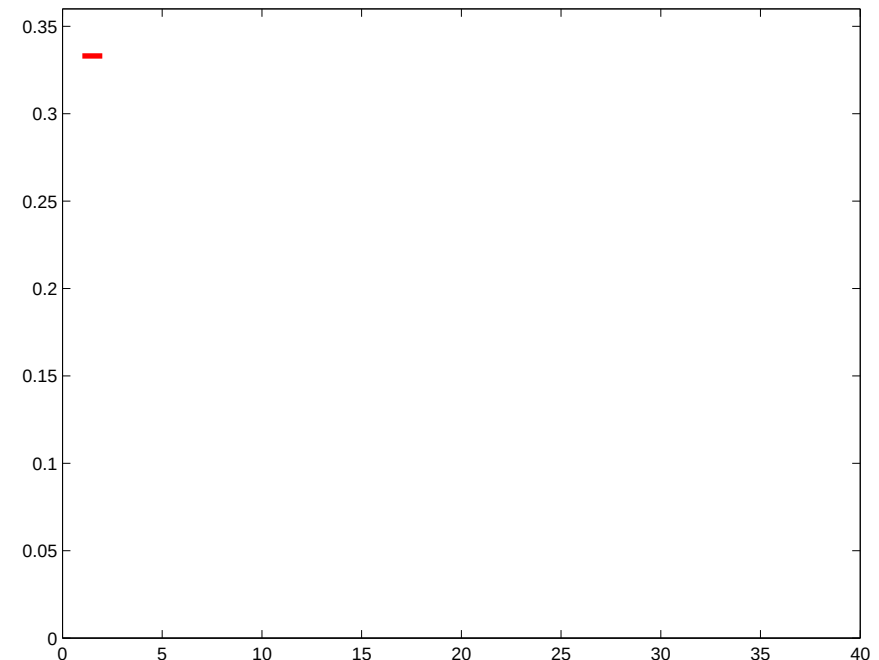
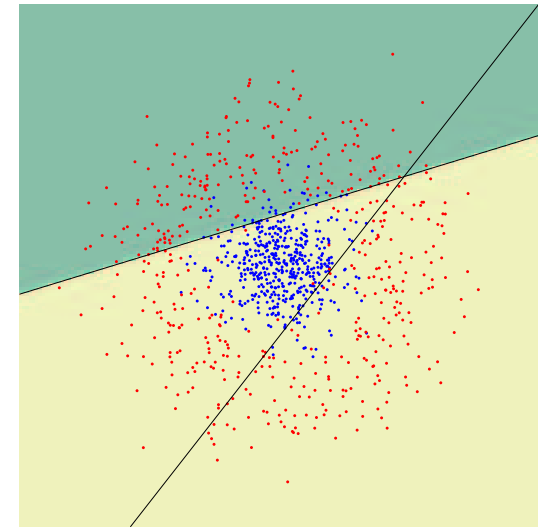
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

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 algorithmus

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

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

For $t = 1, \dots, T$:

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

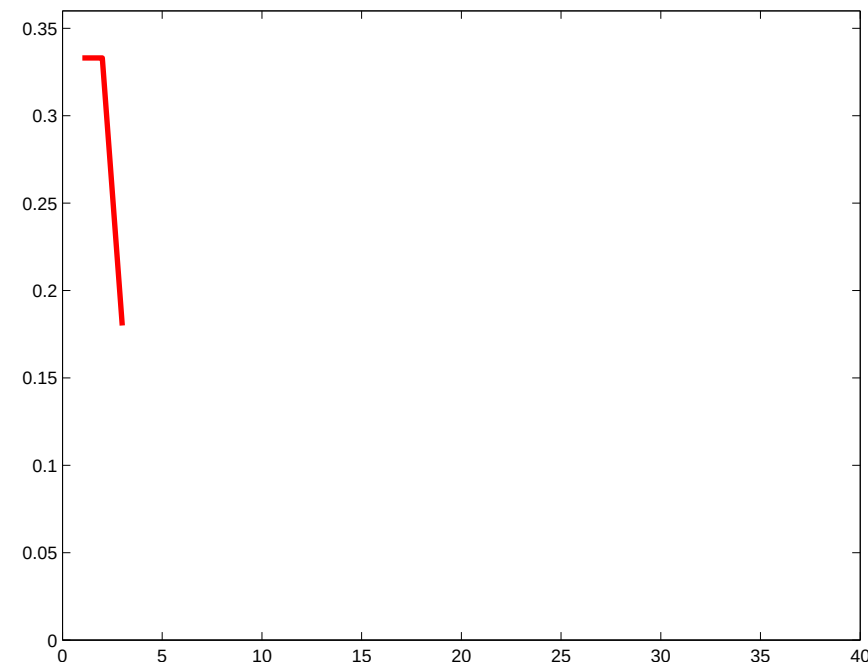
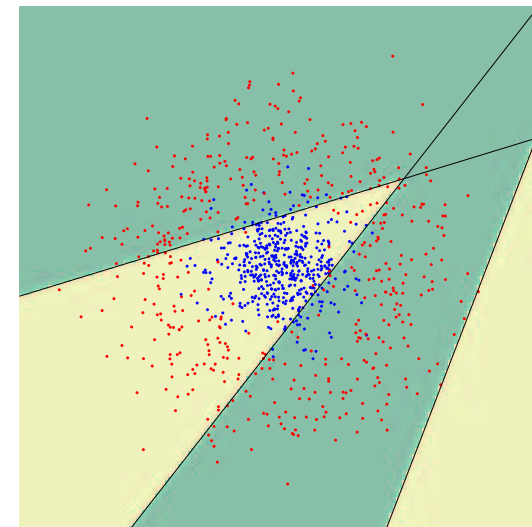
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

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 algorithmus

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

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

For $t = 1, \dots, T$:

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

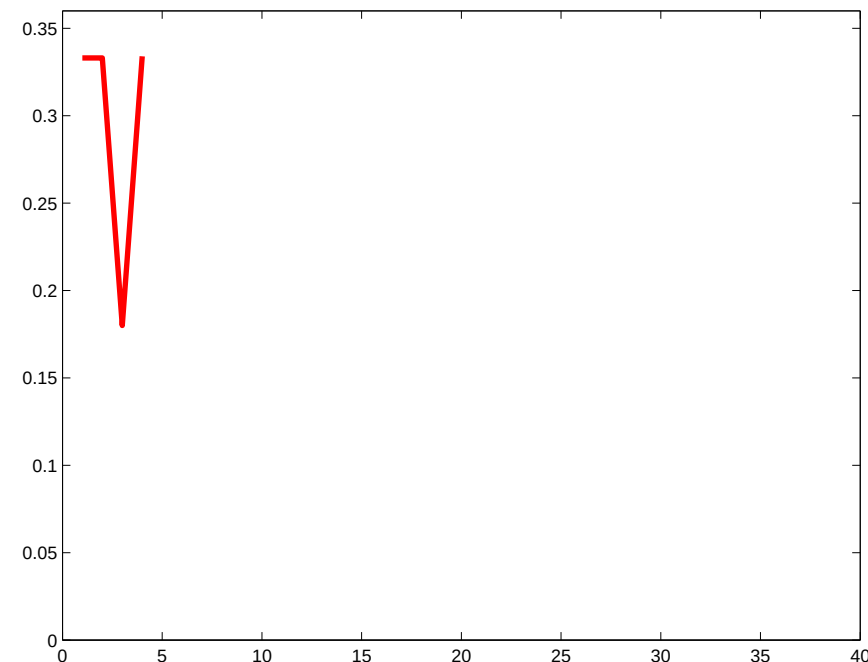
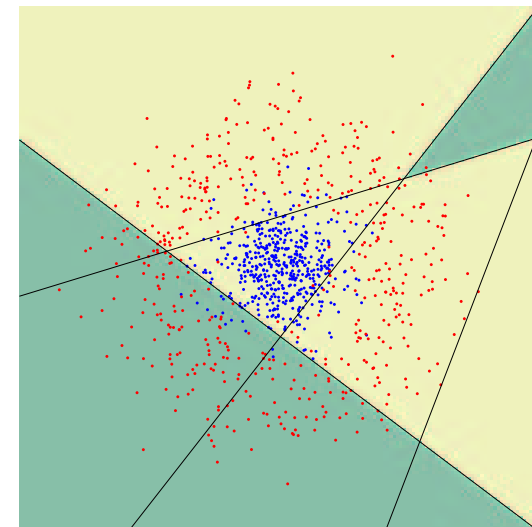
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is normalisation factor

Output the final classifier:

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

$t = 4$



AdaBoost algorithmus

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

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

For $t = 1, \dots, T$:

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

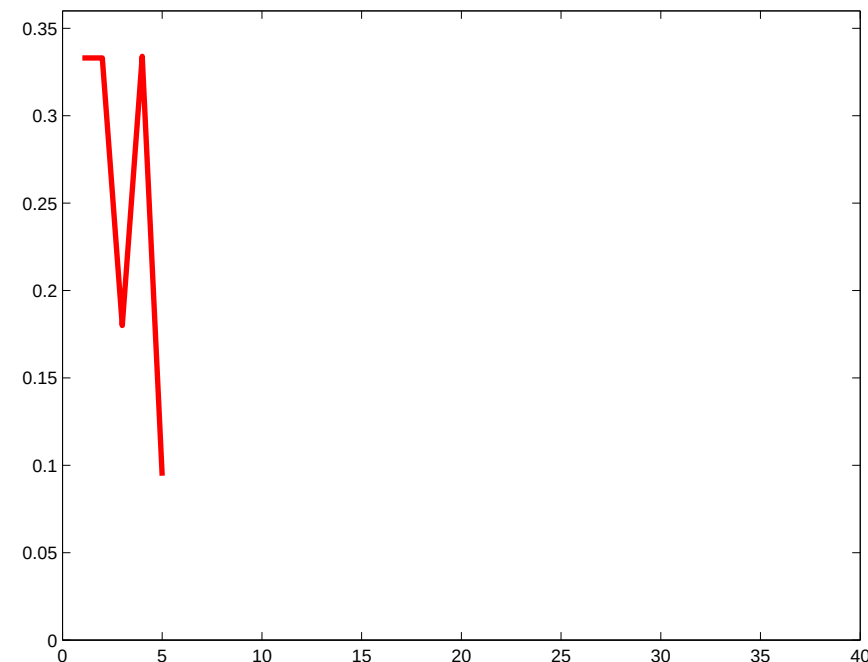
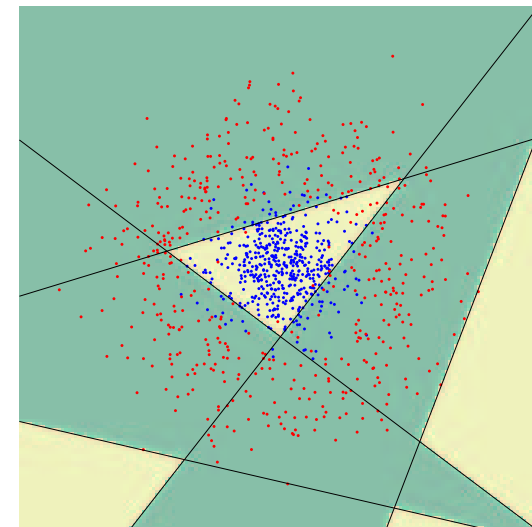
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is normalisation factor

Output the final classifier:

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

$t = 5$



AdaBoost algorithmus

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

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

For $t = 1, \dots, T$:

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

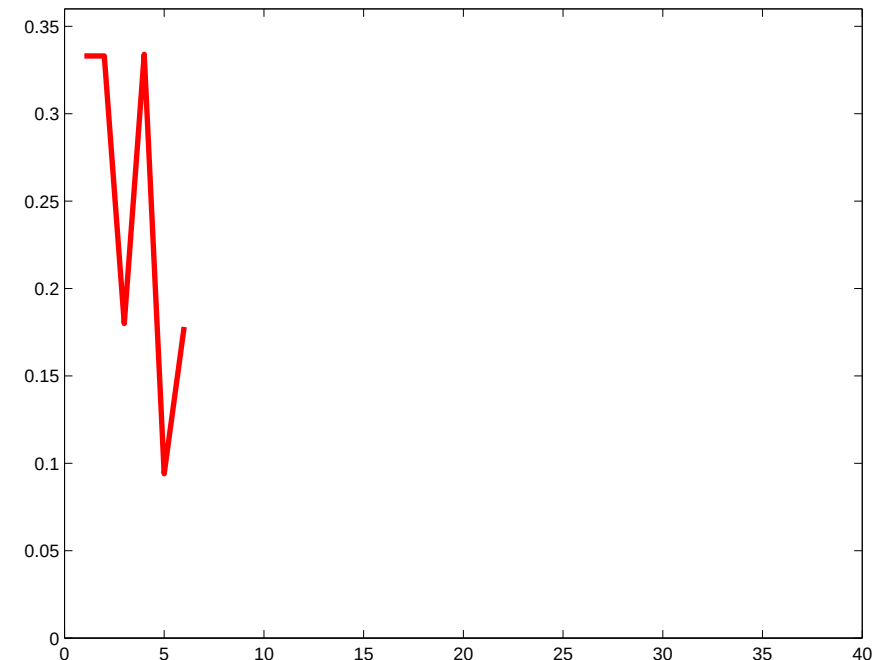
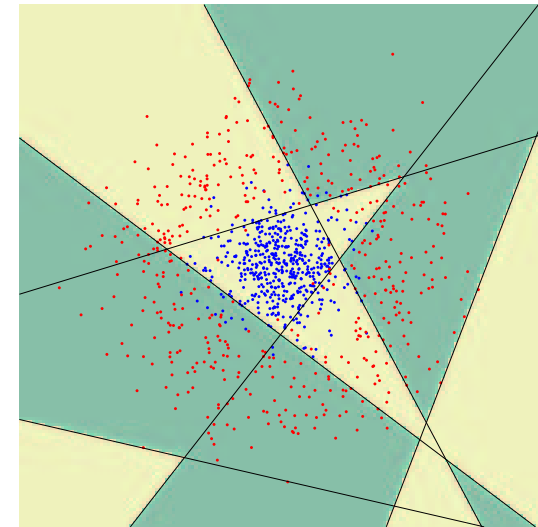
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is normalisation factor

Output the final classifier:

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

$t = 6$



AdaBoost algorithmus

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

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

For $t = 1, \dots, T$:

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

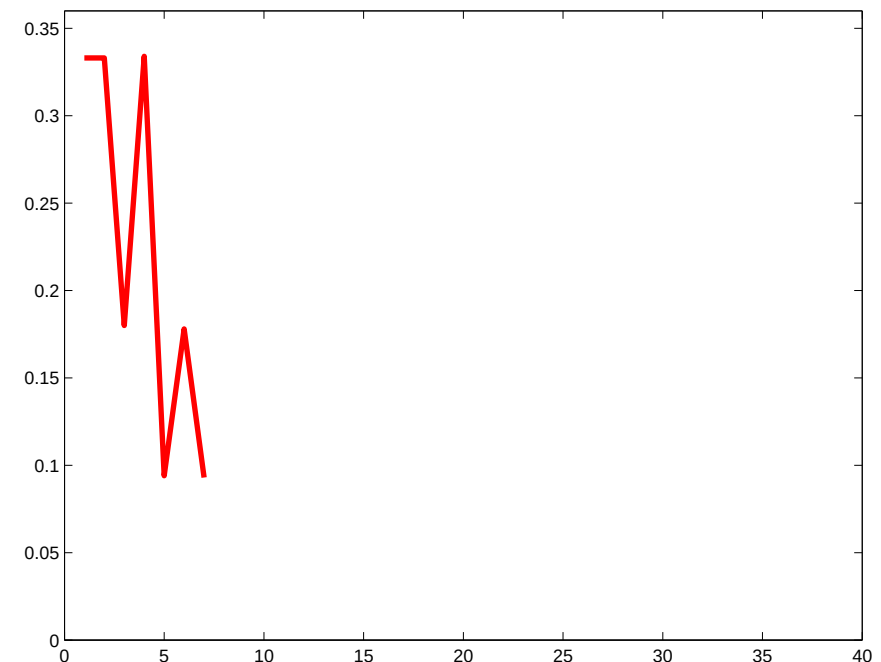
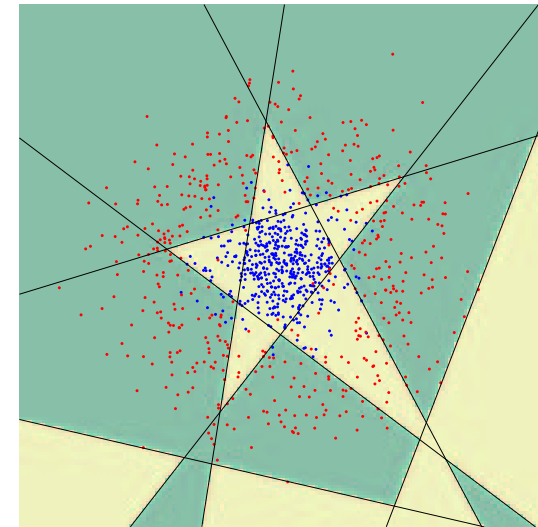
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is normalisation factor

Output the final classifier:

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

$t = 7$



AdaBoost algorithmus

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

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

For $t = 1, \dots, T$:

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

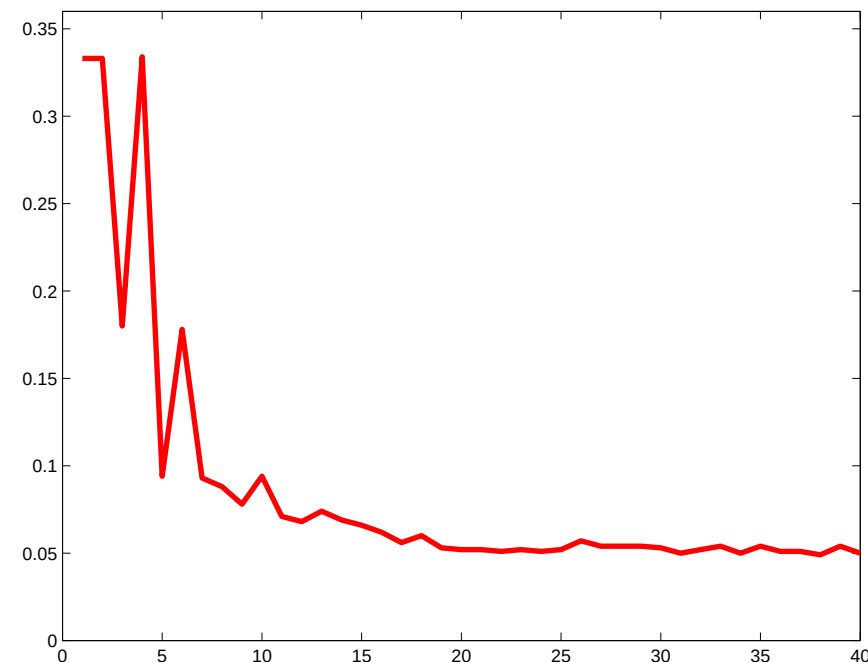
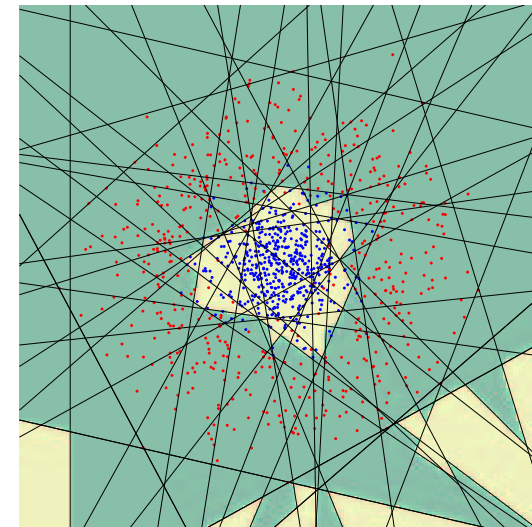
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is normalisation factor

Output the final classifier:

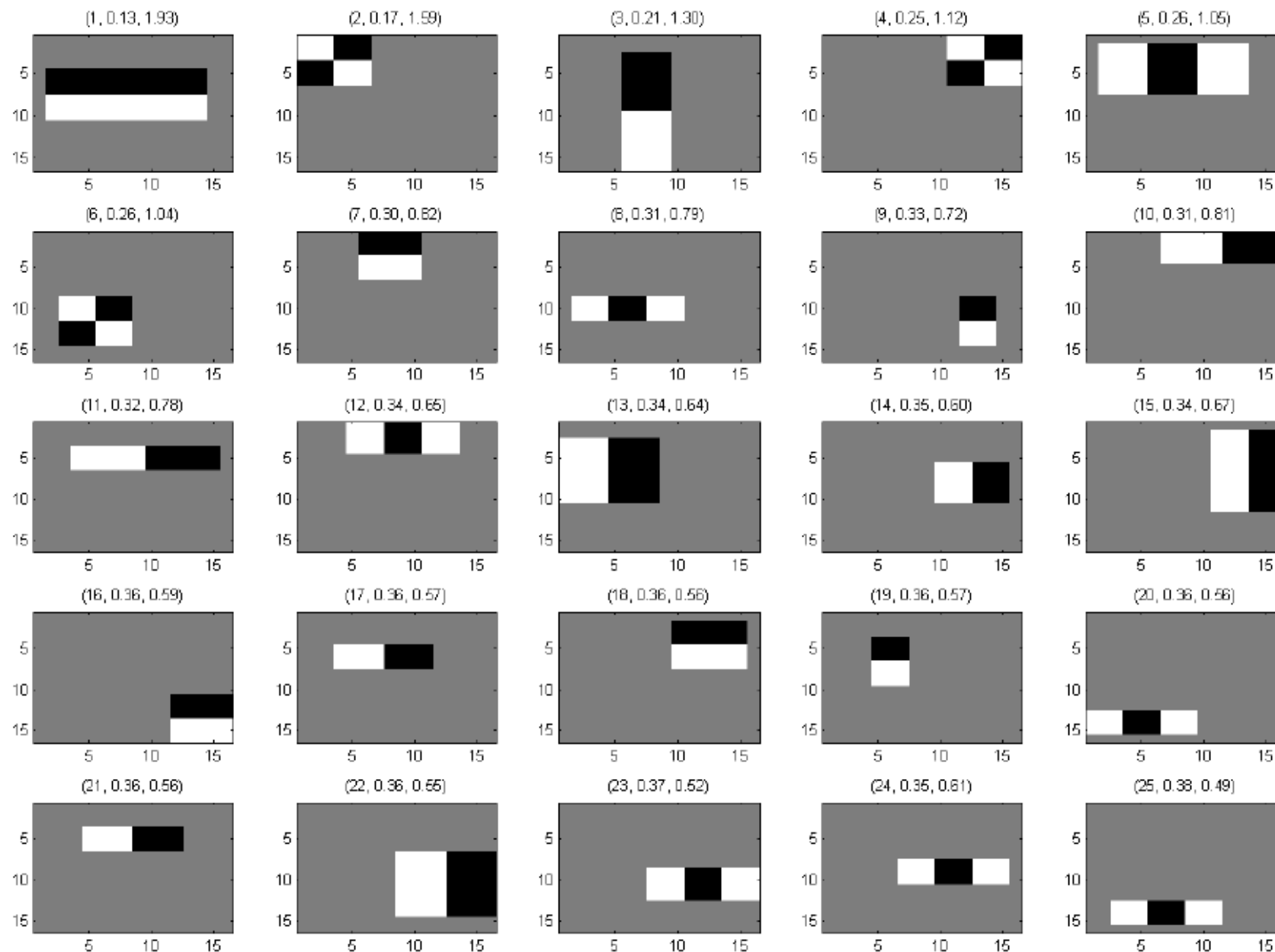
$$H(x) = \text{sign} \left(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \dots + \alpha_{40} h_{40}(x) \right)$$

$t = 40$



Příklady nalezených slabých klasifikátorů pomocí AdaBoostu

Po prvních 25 iteracích.



Detektor tváří: vývoj

- ◆ A. Wald. Sequential analysis. Dover, New York, 1947.
 - Nástroj pro kontrolu kvality průmyslové výroby za 2. světové války.

Detektor tváří: vývoj

- ◆ A. Wald. Sequential analysis. Dover, New York, 1947.
 - Nástroj pro kontrolu kvality průmyslové výroby za 2. světové války.
- ◆ Y. Freund, R. E. Schapire: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences. 1997.
 - AdaBoost algoritmus: sestroj silný klasifikátor ze slabých.

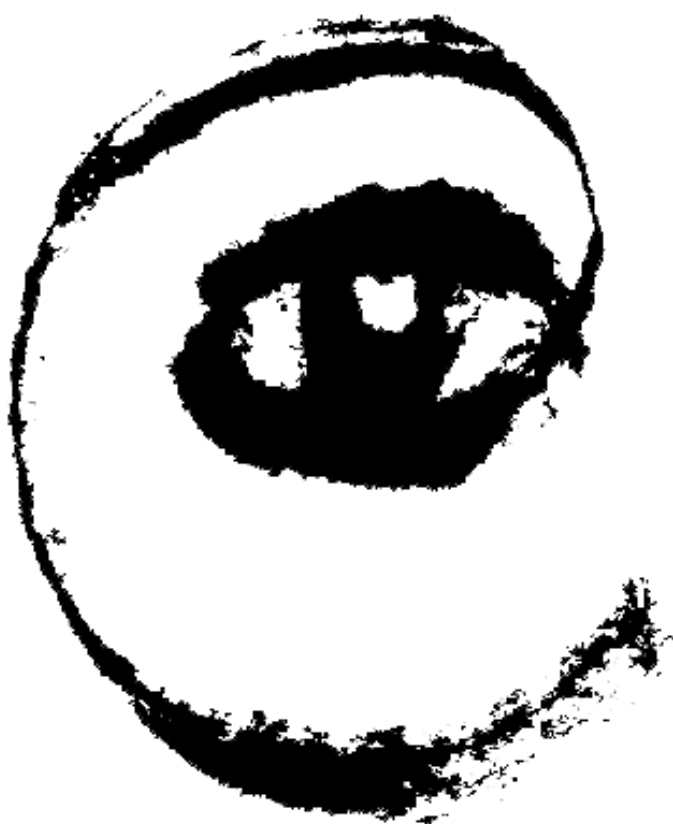
Detektor tváří: vývoj

- ◆ A. Wald. Sequential analysis. Dover, New York, 1947.
 - Nástroj pro kontrolu kvality průmyslové výroby za 2. světové války.
- ◆ Y. Freund, R. E. Schapire: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences. 1997.
 - AdaBoost algoritmus: sestroj silný klasifikátor ze slabých.
- ◆ P. Viola, M. Jones: Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition, 2001.
 - Průlom v detekci objektů v reálném čase.
 - článek má 29,099 citací (google scholar, Nov 2021).

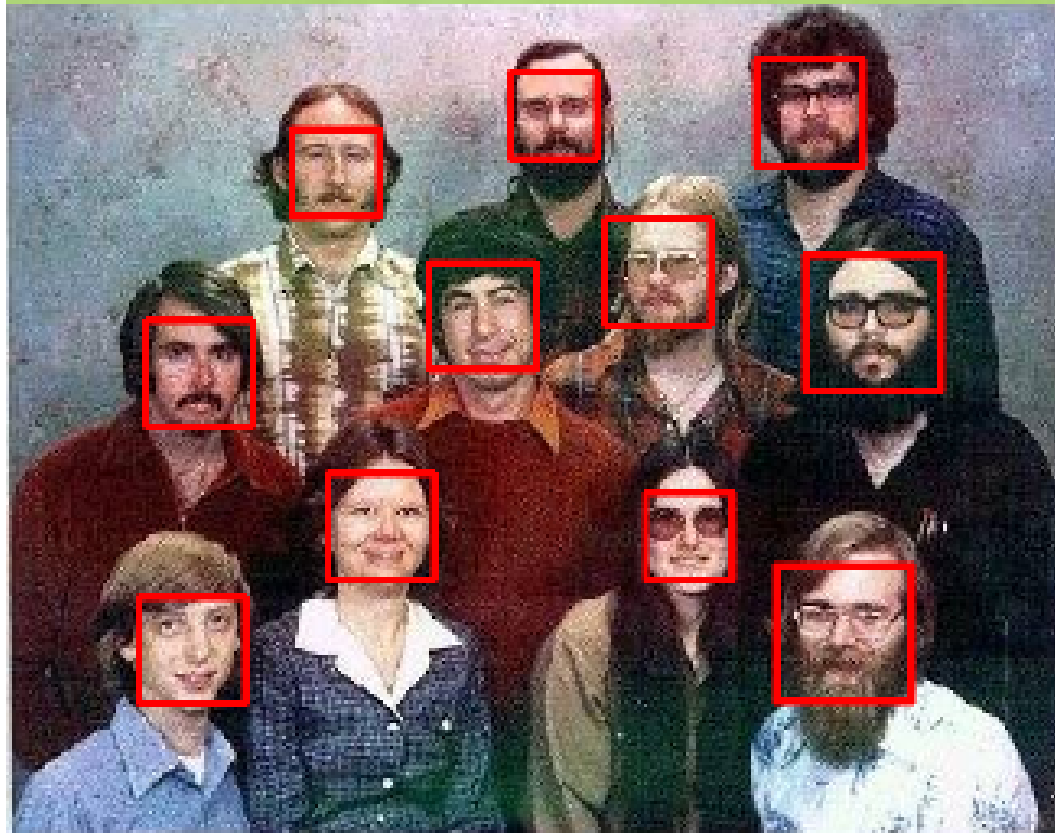
Detektor tváří: vývoj

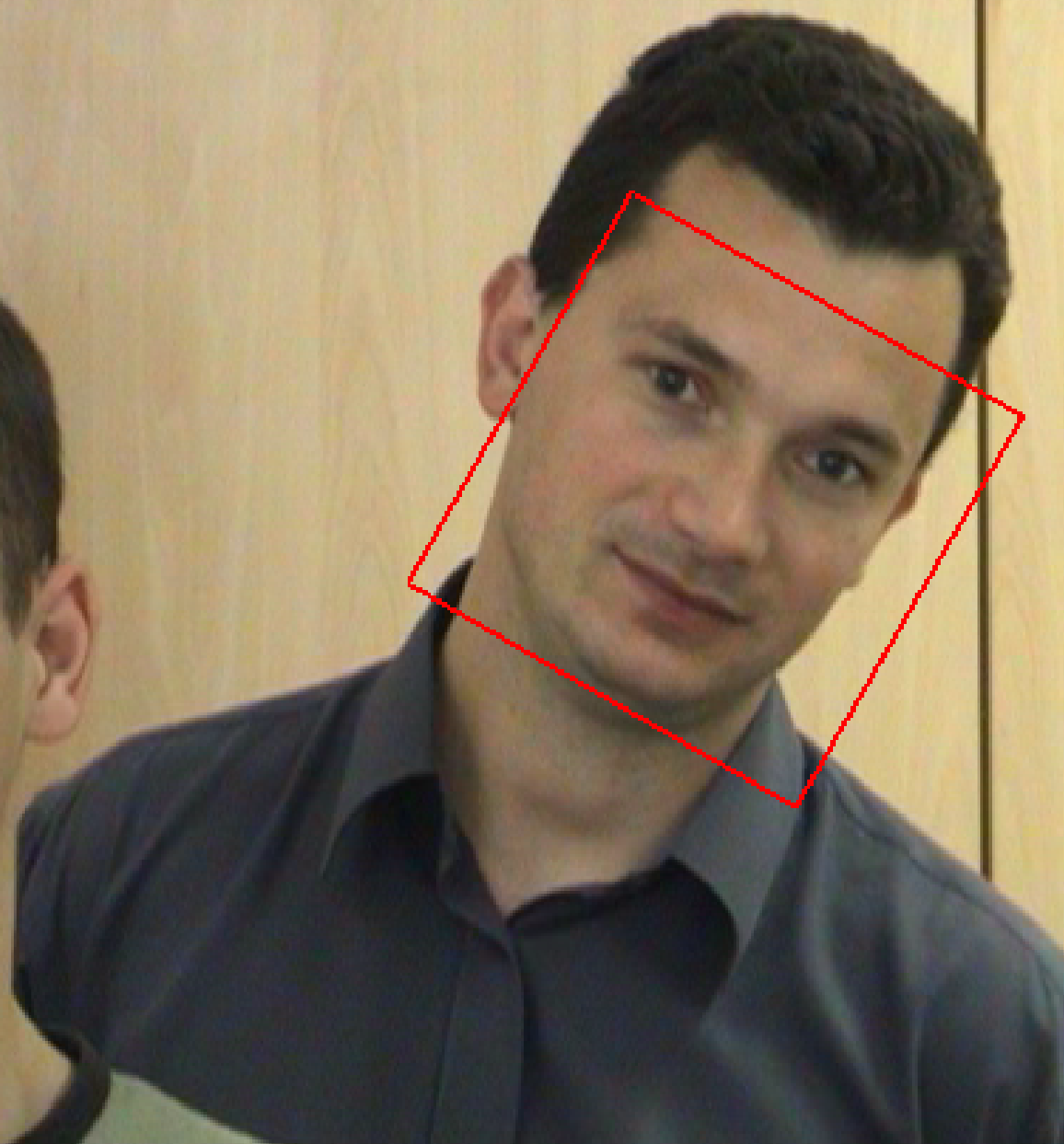
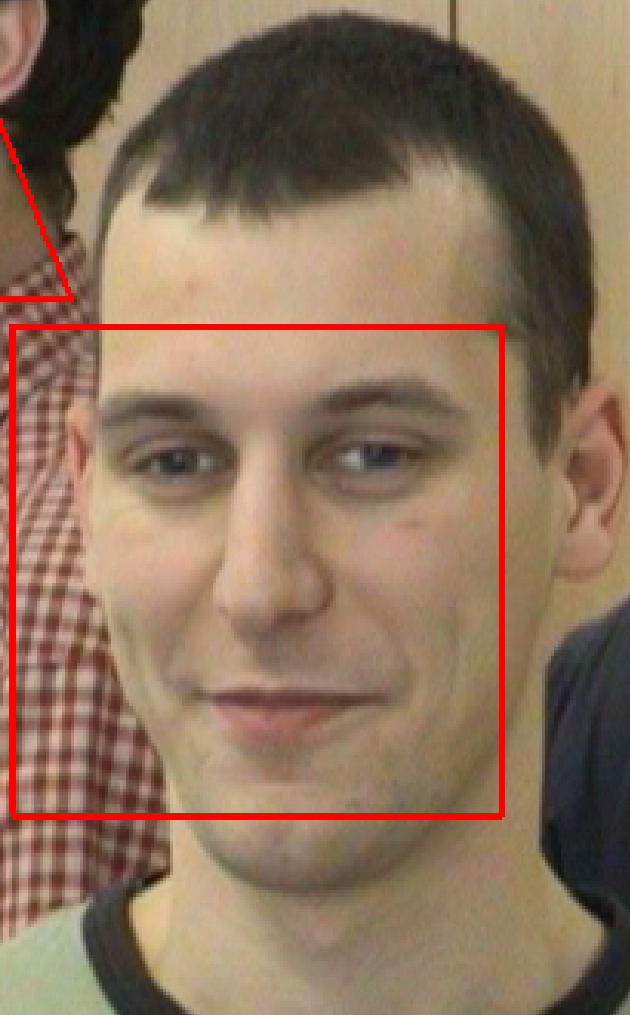
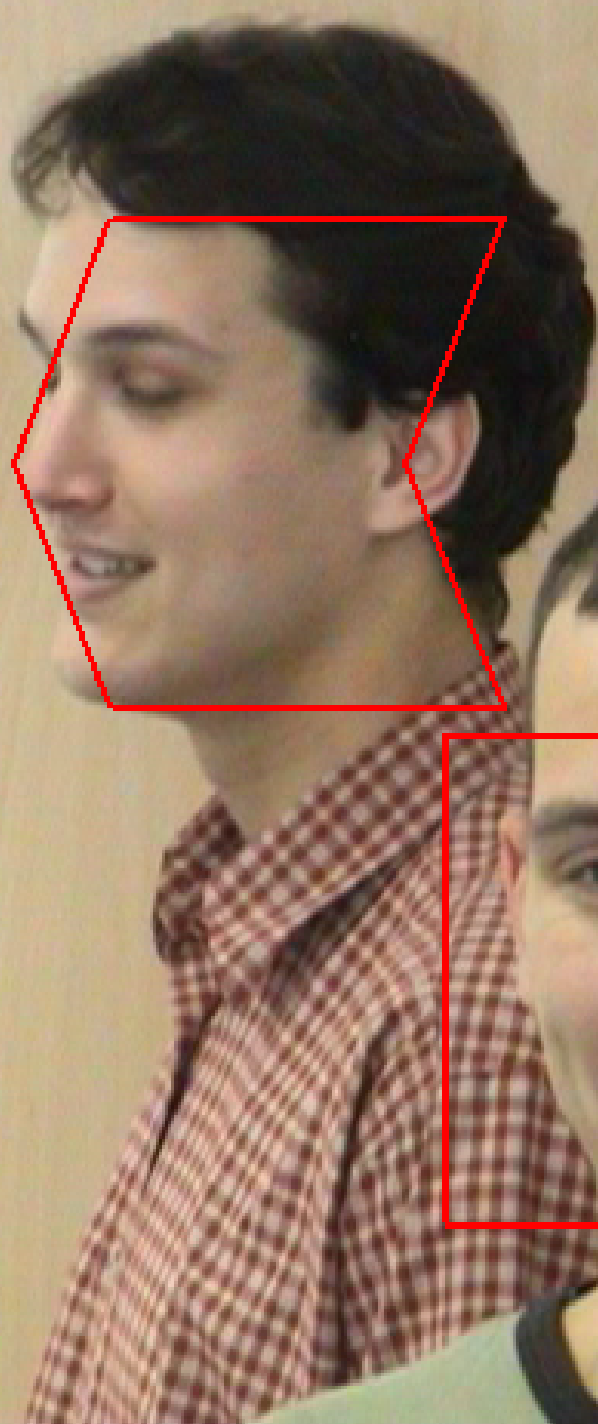
- ◆ A. Wald. Sequential analysis. Dover, New York, 1947.
 - Nástroj pro kontrolu kvality průmyslové výroby za 2. světové války.
- ◆ Y. Freund, R. E. Schapire: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences. 1997.
 - AdaBoost algoritmus: sestroj silný klasifikátor ze slabých.
- ◆ P. Viola, M. Jones: Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition, 2001.
 - Průlom v detekci objektů v reálném čase.
 - článek má 29,099 citací (google scholar, Nov 2021).
- ◆ J. Sochman, J. Matas: Waldboost-learning for time constrained sequential detection. Computer Vision and Pattern Recognition, 2005.
 - Optimání nastavení parametrů sekvenčního klasifikátoru.
 - Komerční multi-view detektor Eyedea Recognition s.r.o (www.eyedea.cz).

Konec

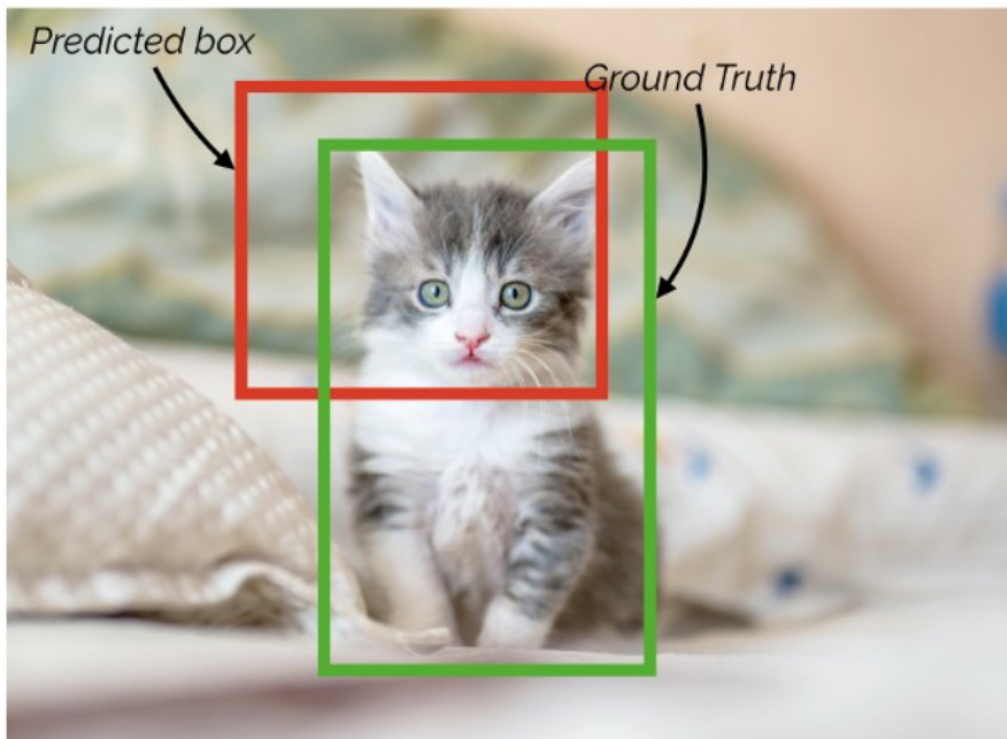


m p



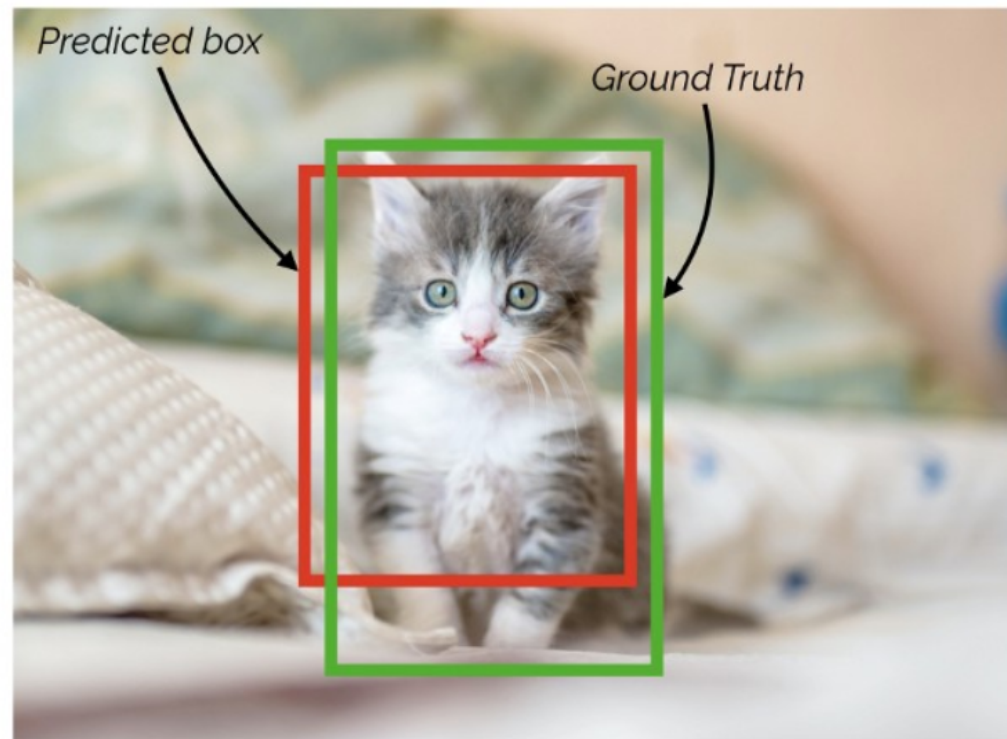


False Positive (FP)

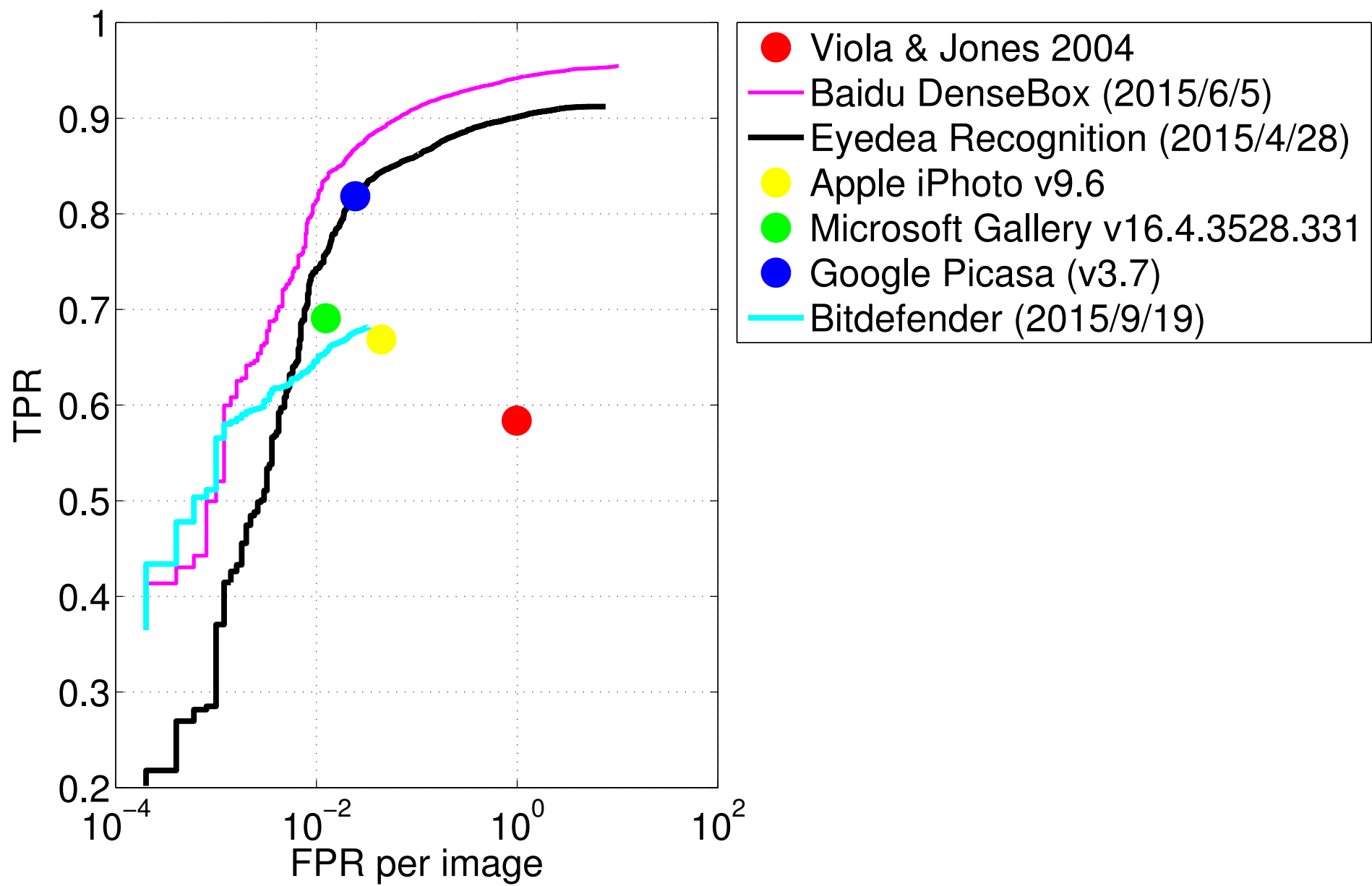


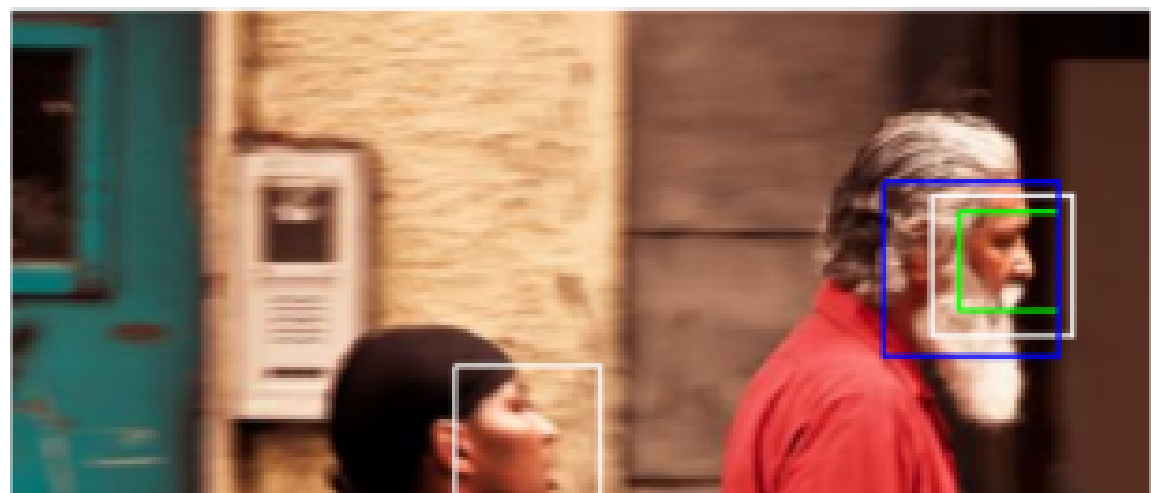
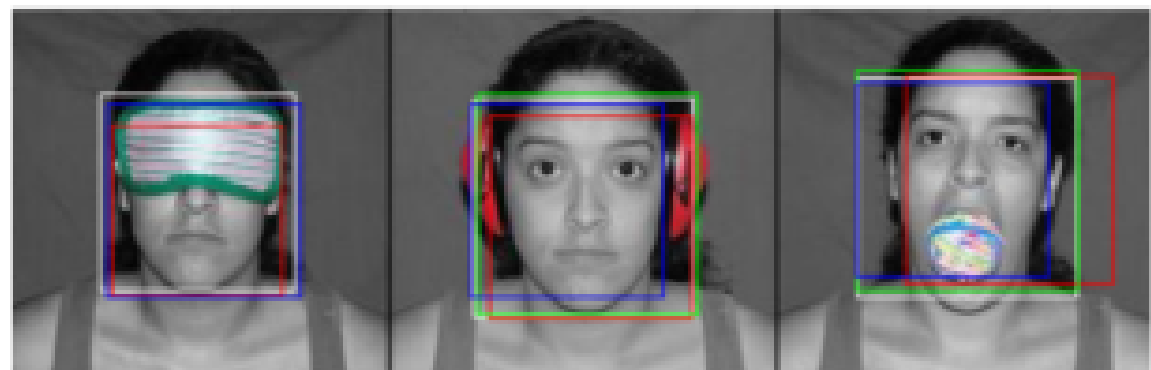
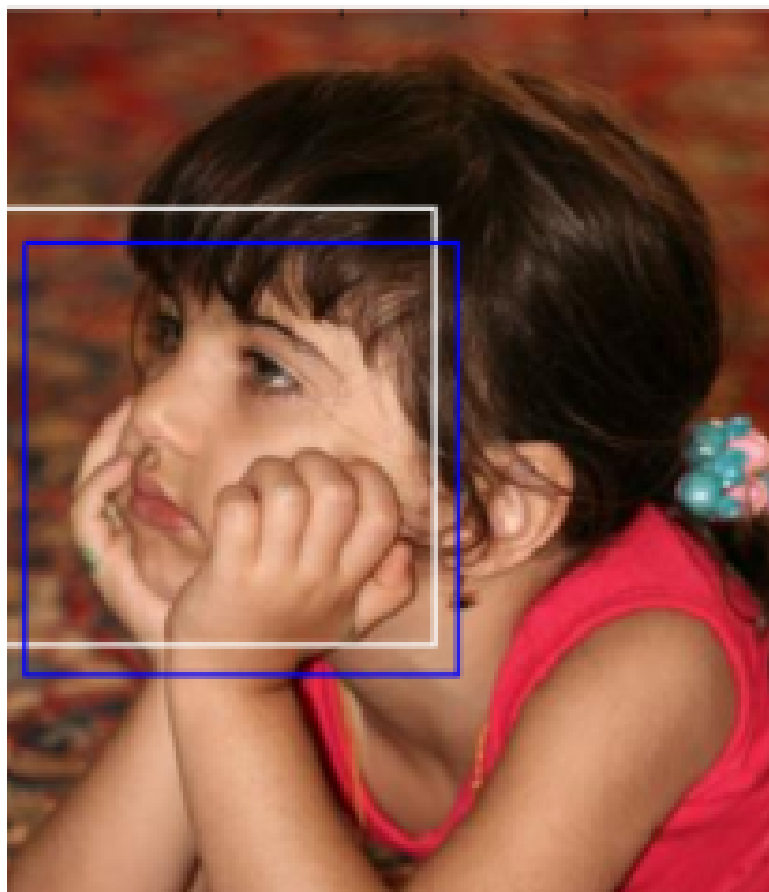
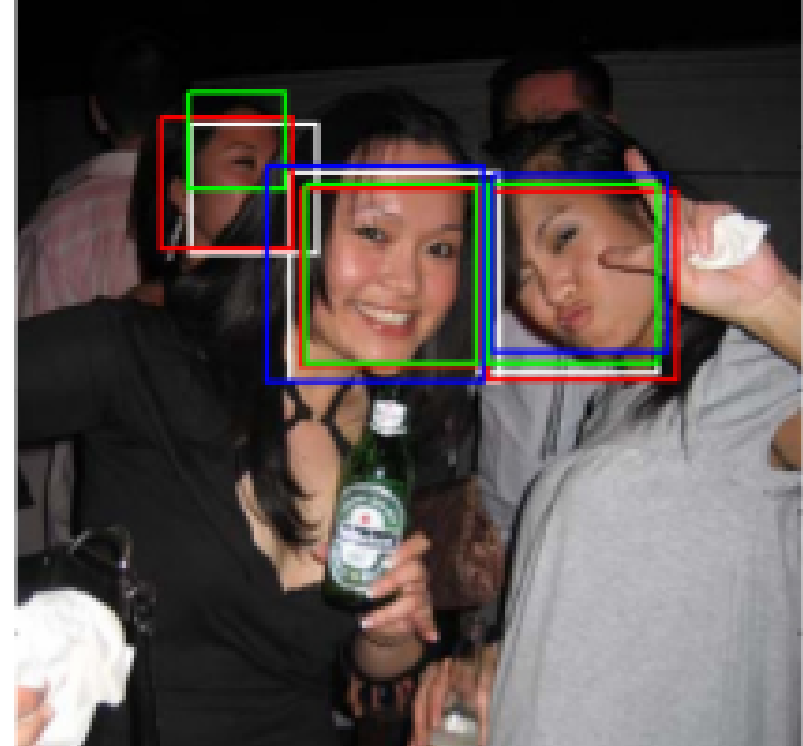
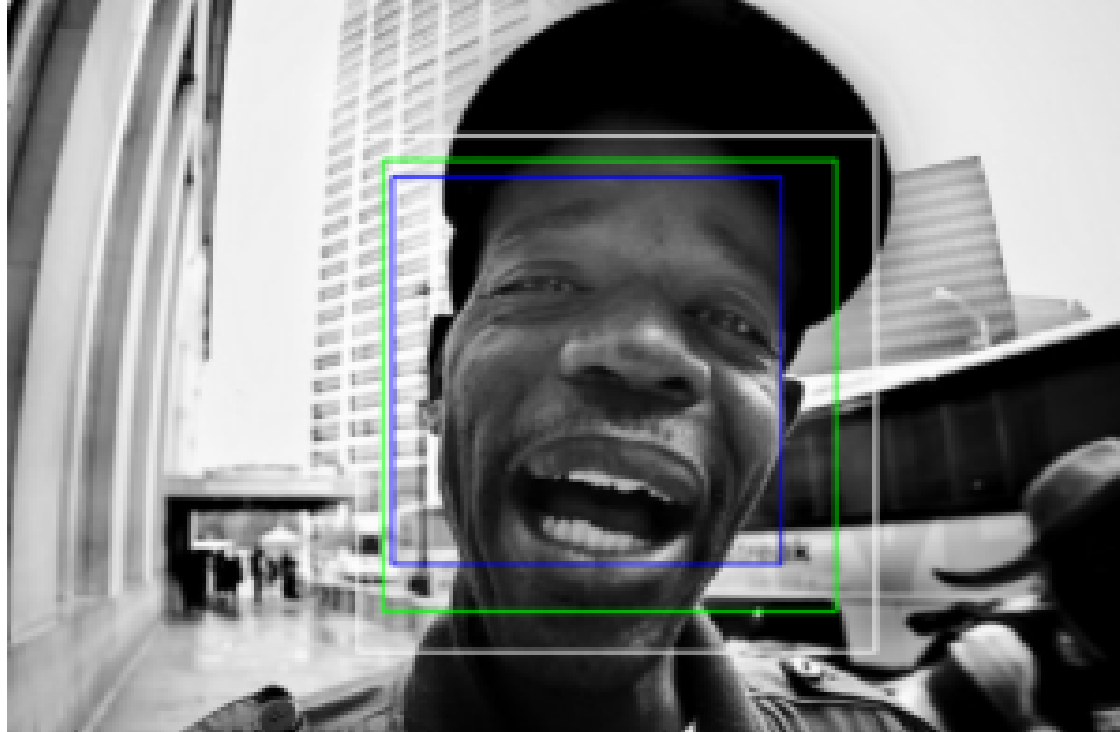
$IoU = \sim 0.3$

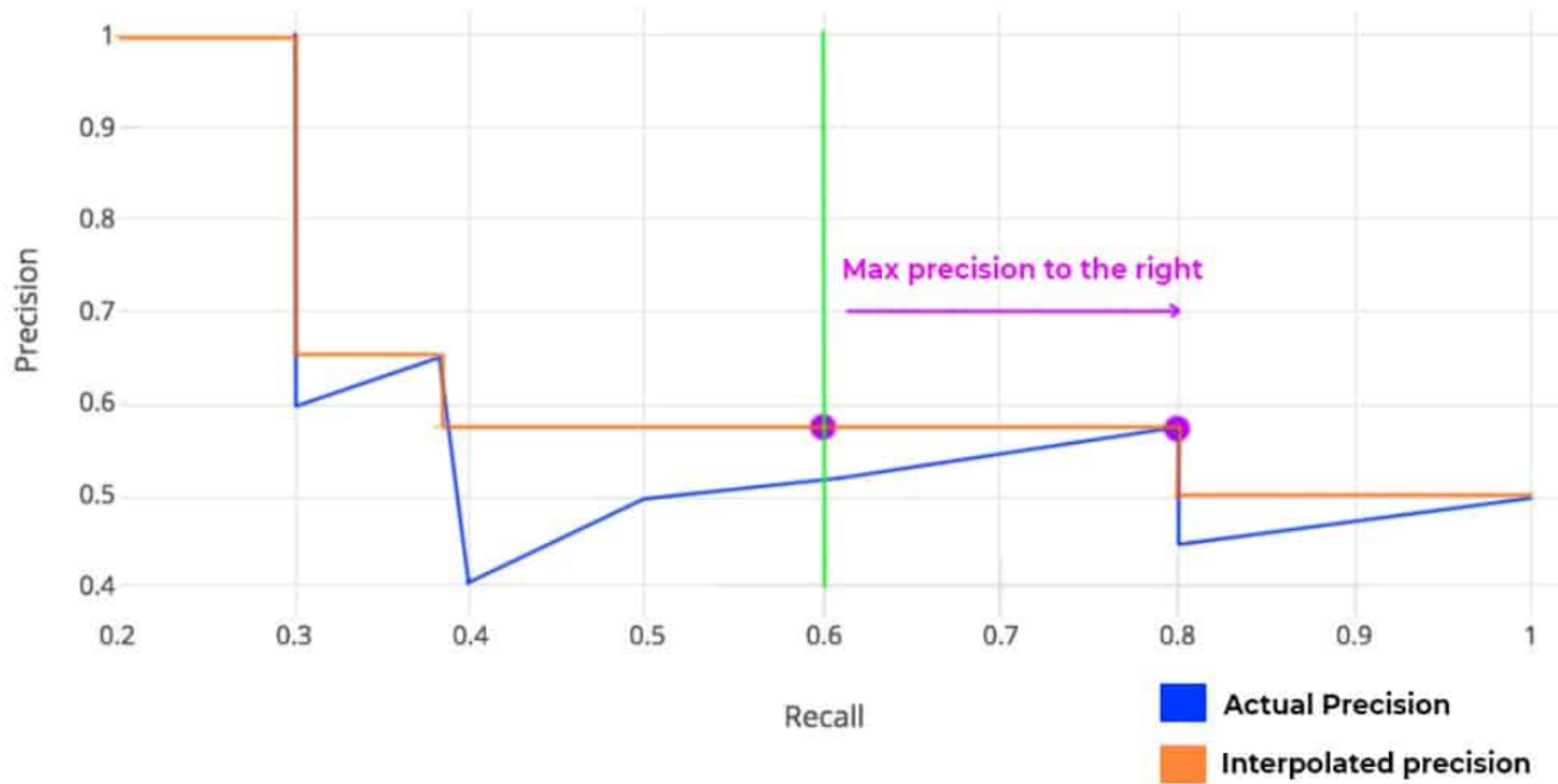
True Positive (TP)



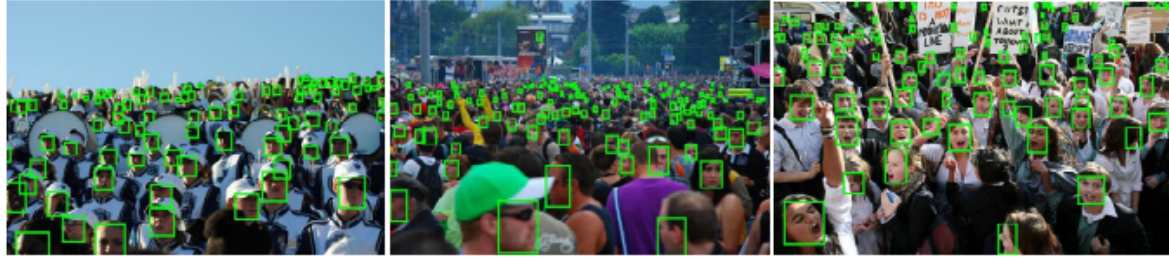
$IoU = \sim 0.7$







Scale



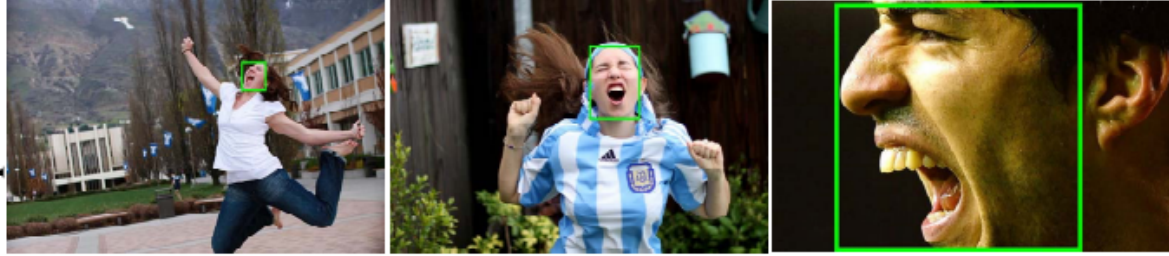
Pose



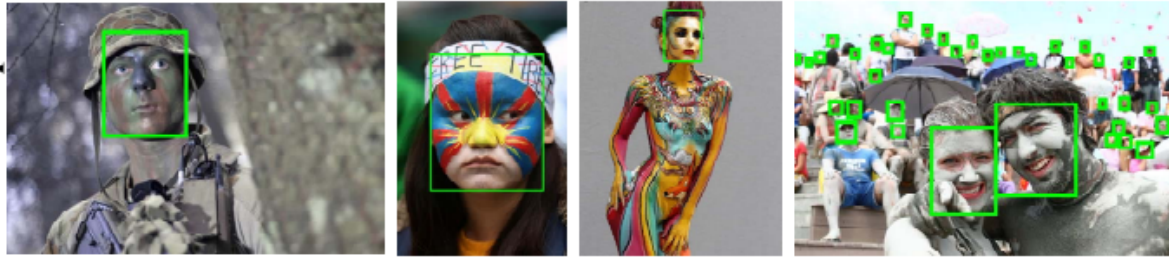
Occlusion



Expression

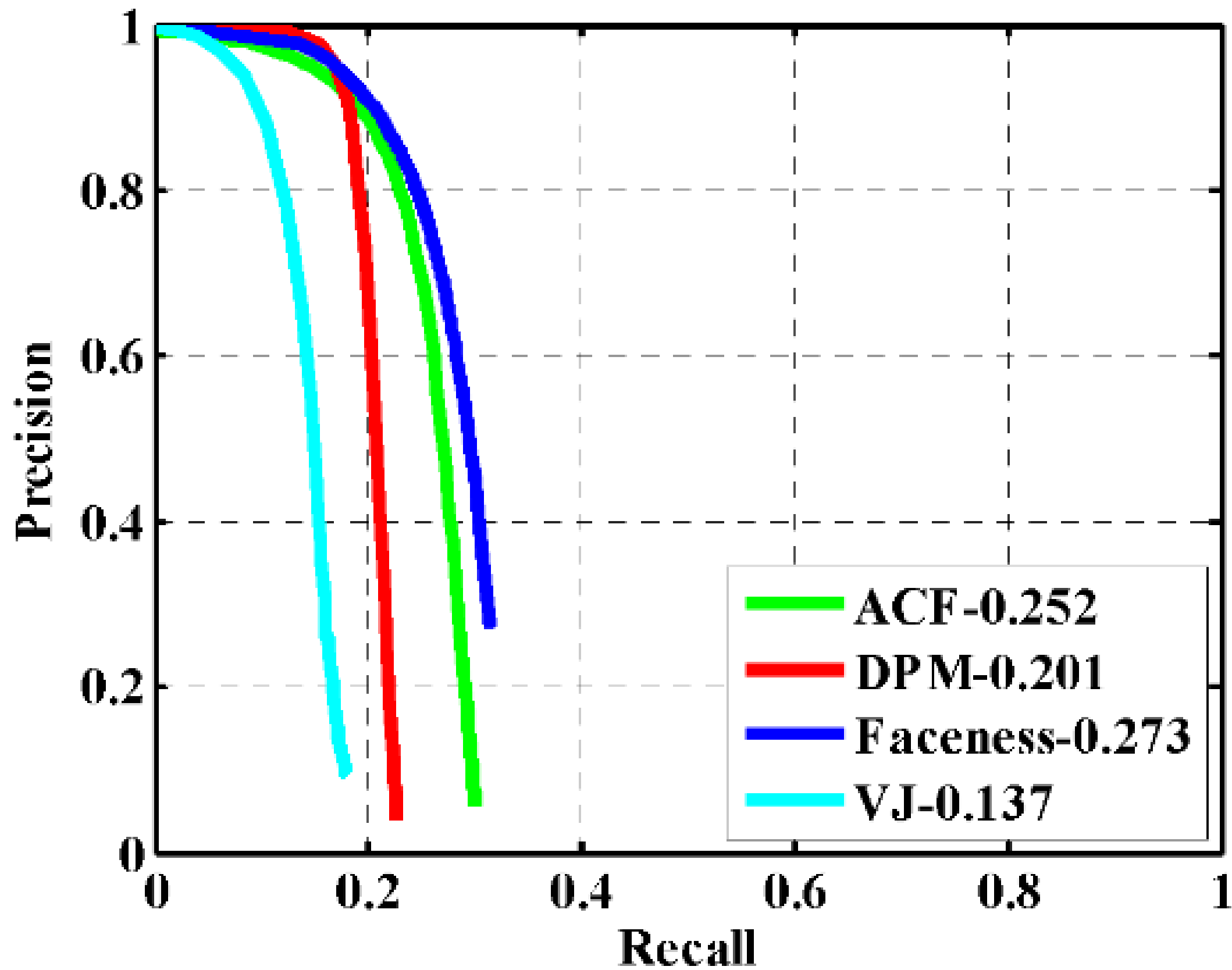


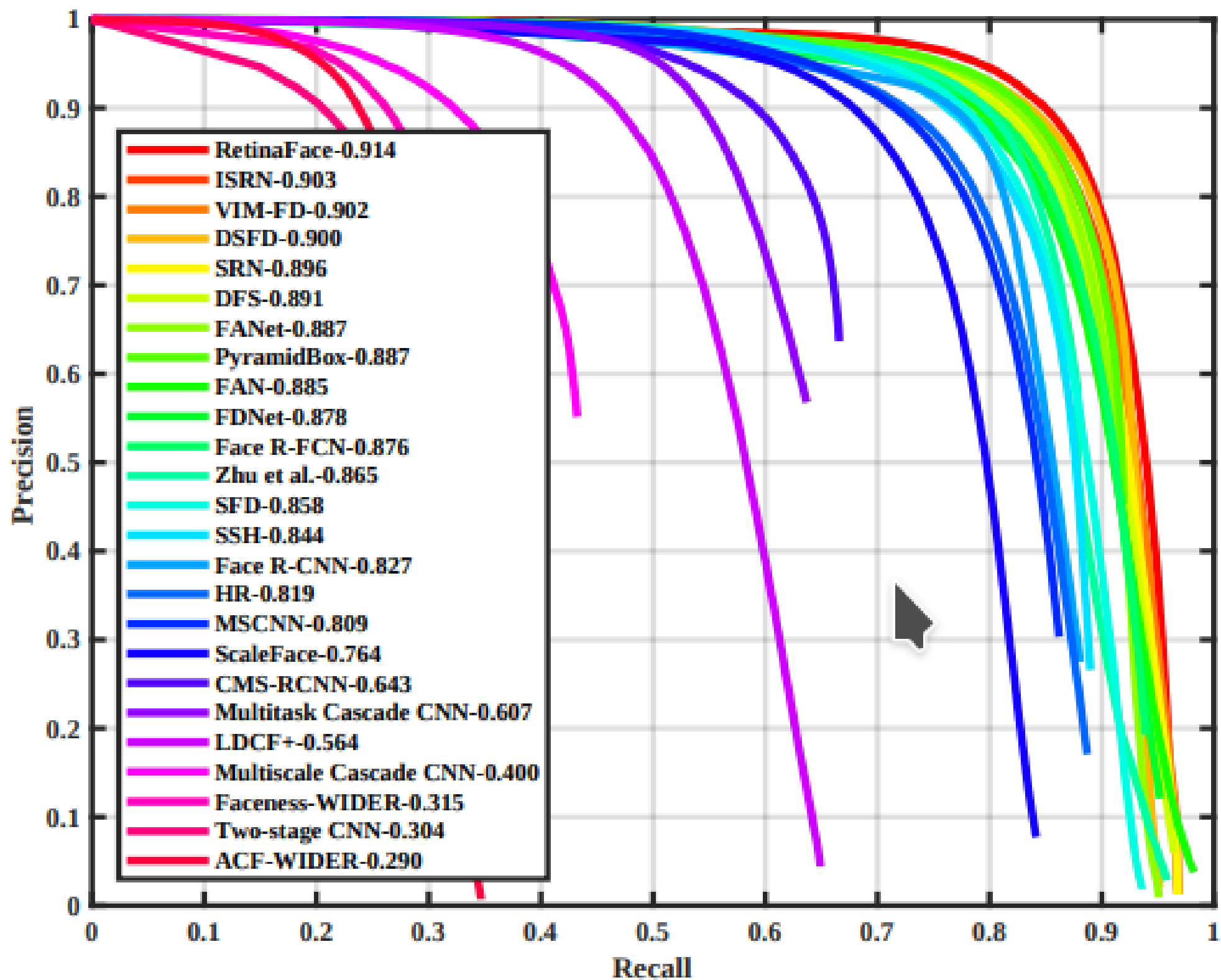
Makeup



Illumination







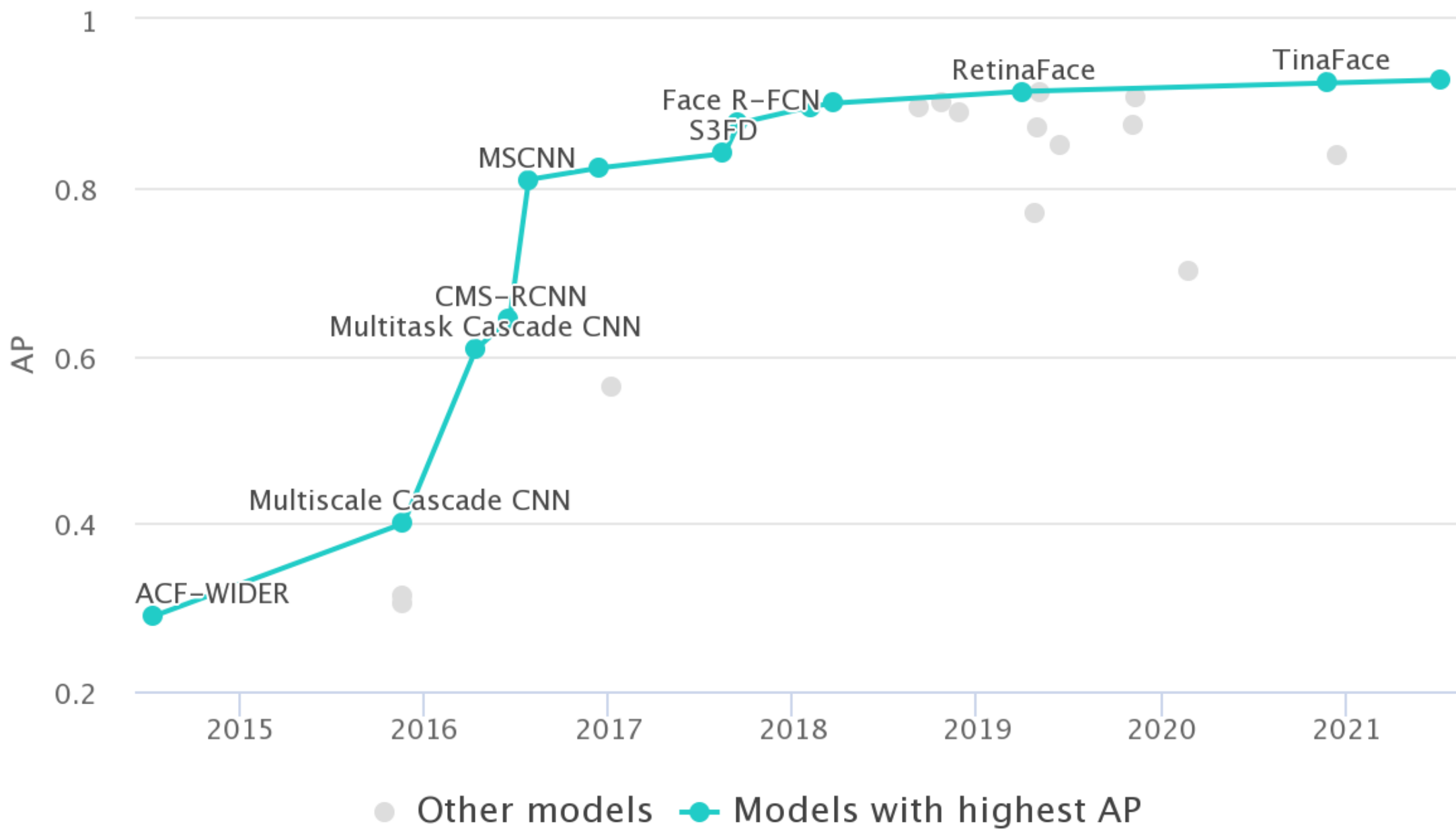


Image A

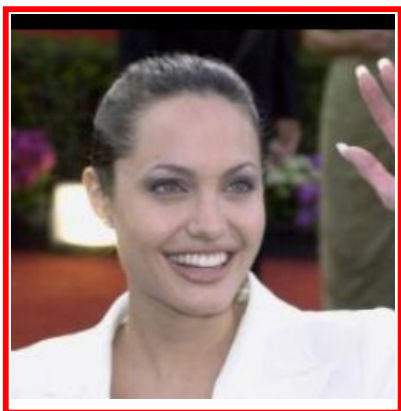
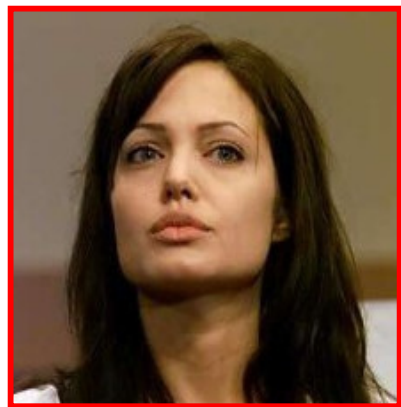


Image B



Verification
system

the same / different

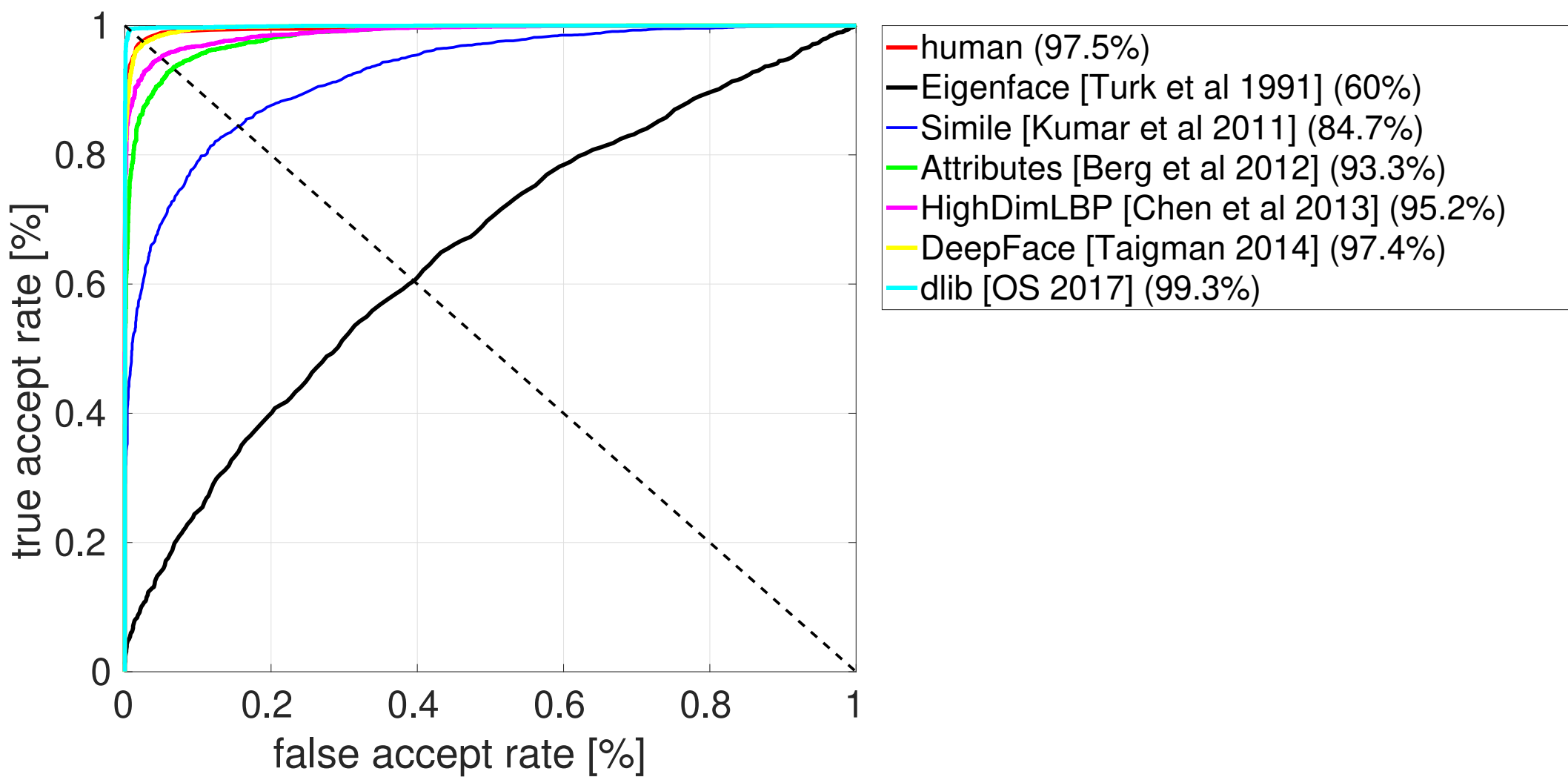


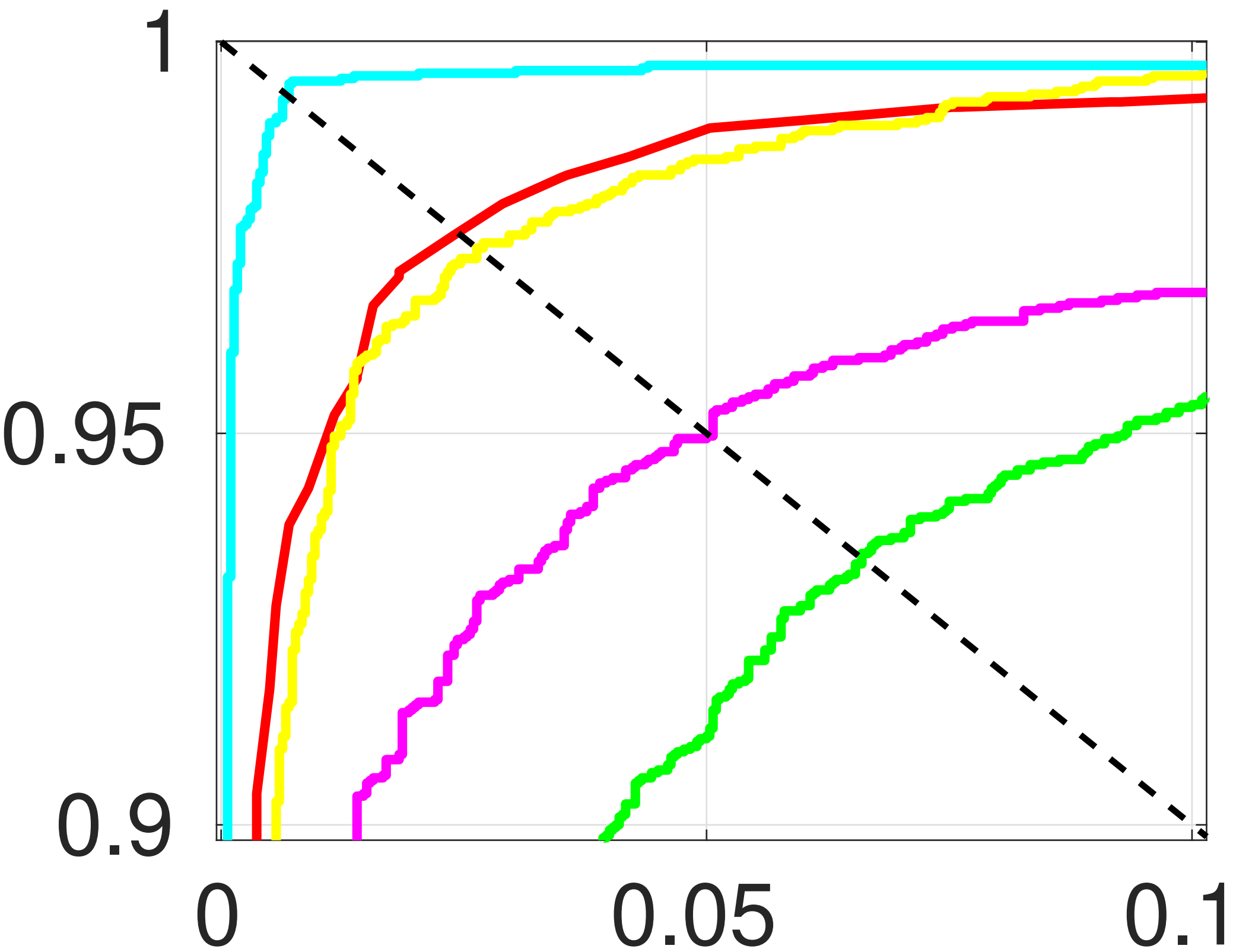




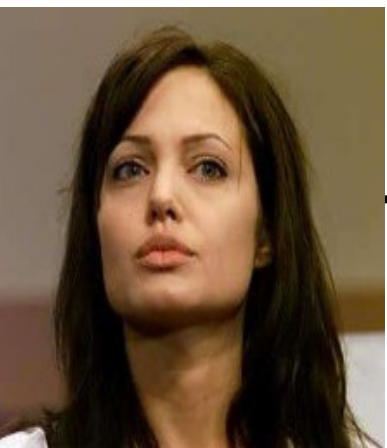








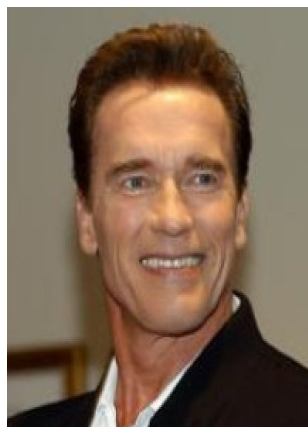
probe



Face recognition
engine

sorted list of N
IDs in gallery

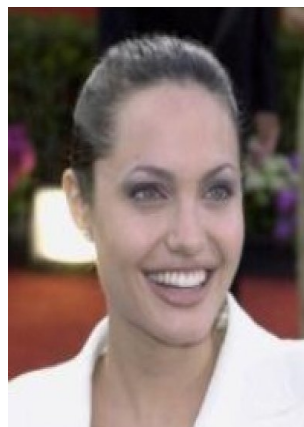
gallery images



ID1



ID2



ID3



ID4

...



IDn



(a) Good quality mugshot

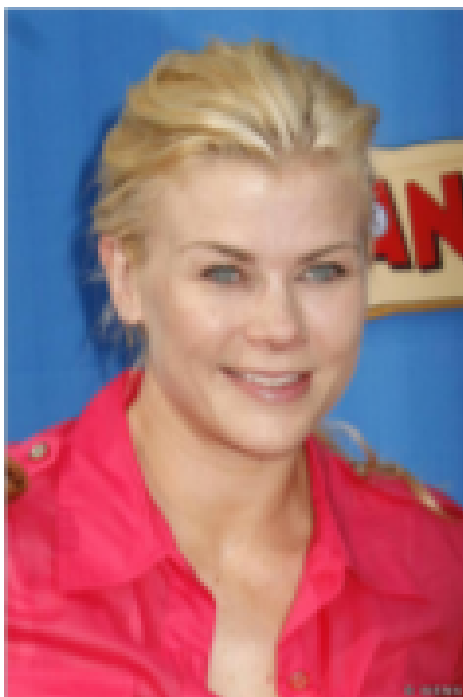


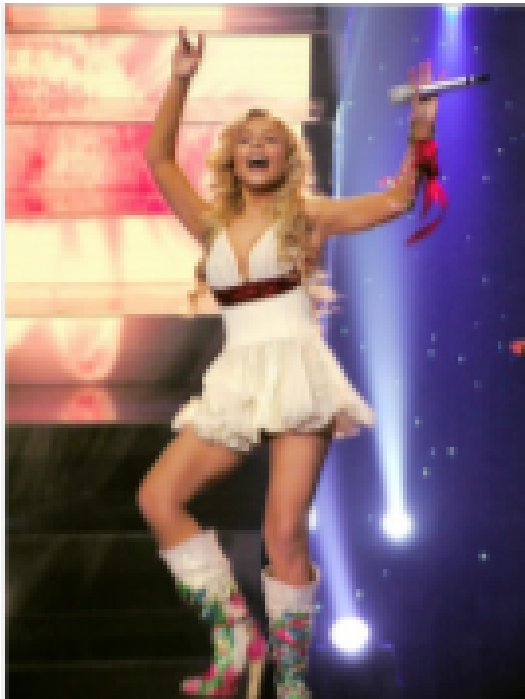
(b) Poor quality webcam





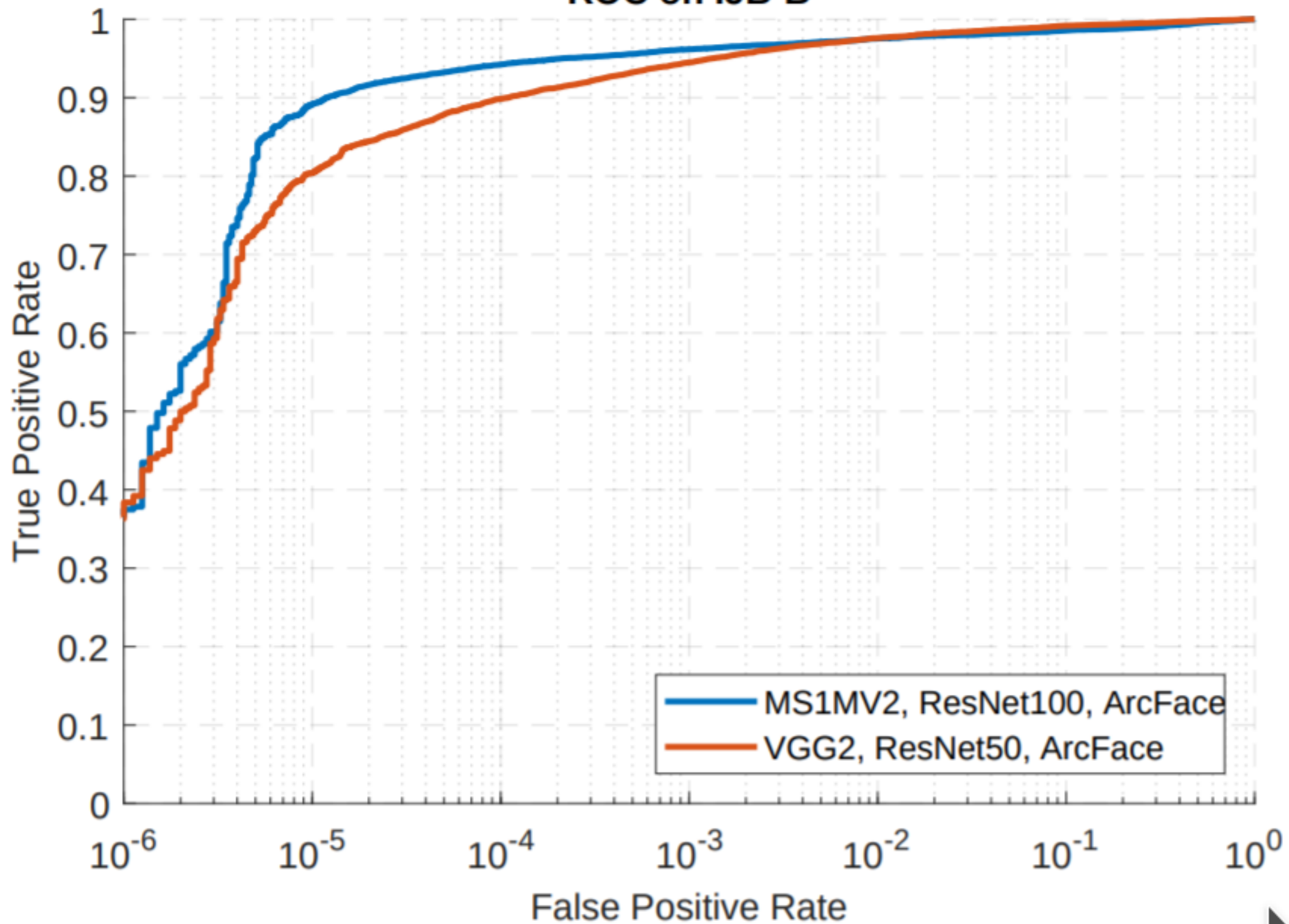








ROC on IJB-B



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.



Take a selfie



Make sure that your face is in the frame and clearly visible.



! **Verify age**

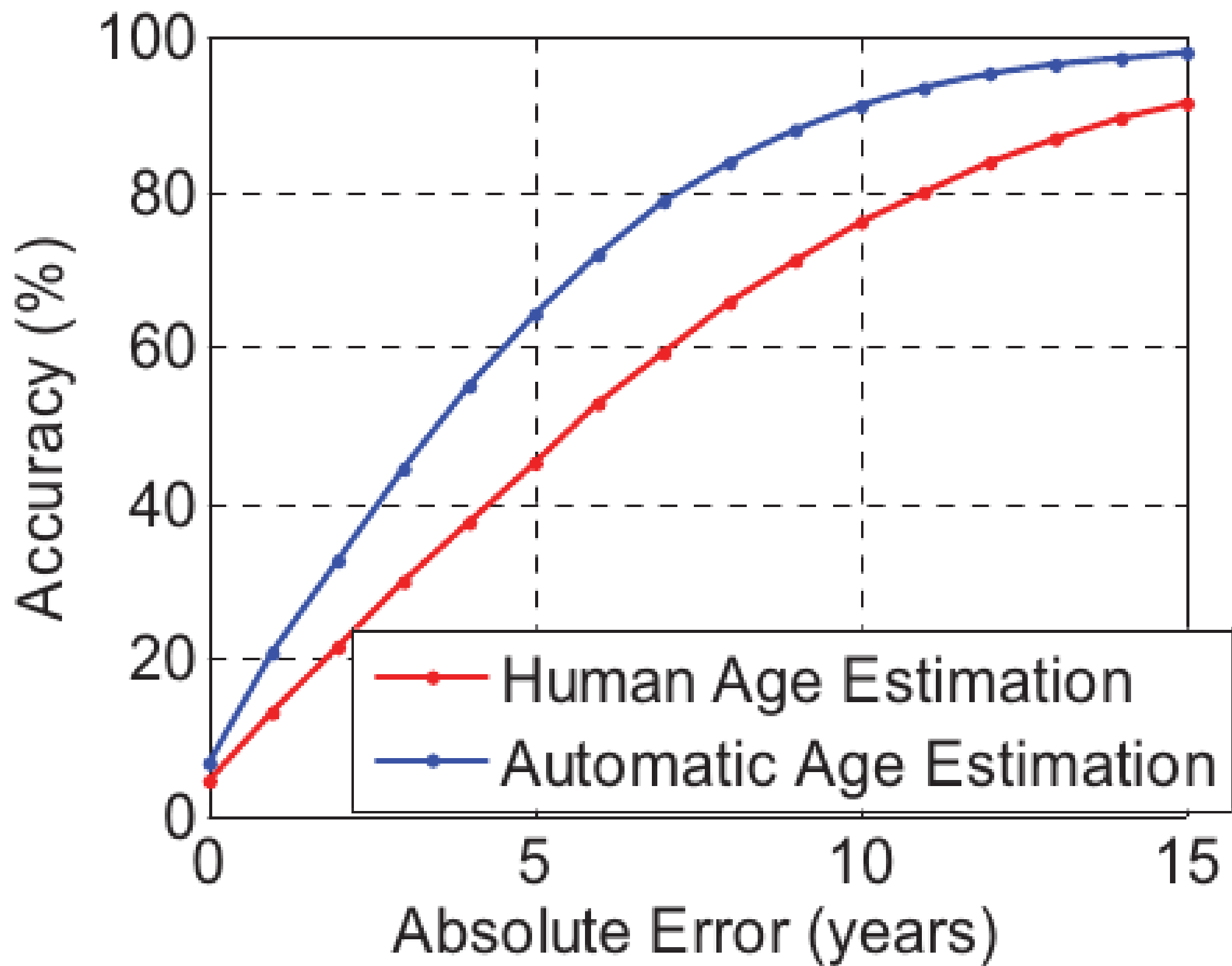
! **Take a selfie**

- ✓ Make sure to be in clear lighting
- ✓ Keep a neutral background
- ✓ Stay within the frame of the photo

✓ Age estimated

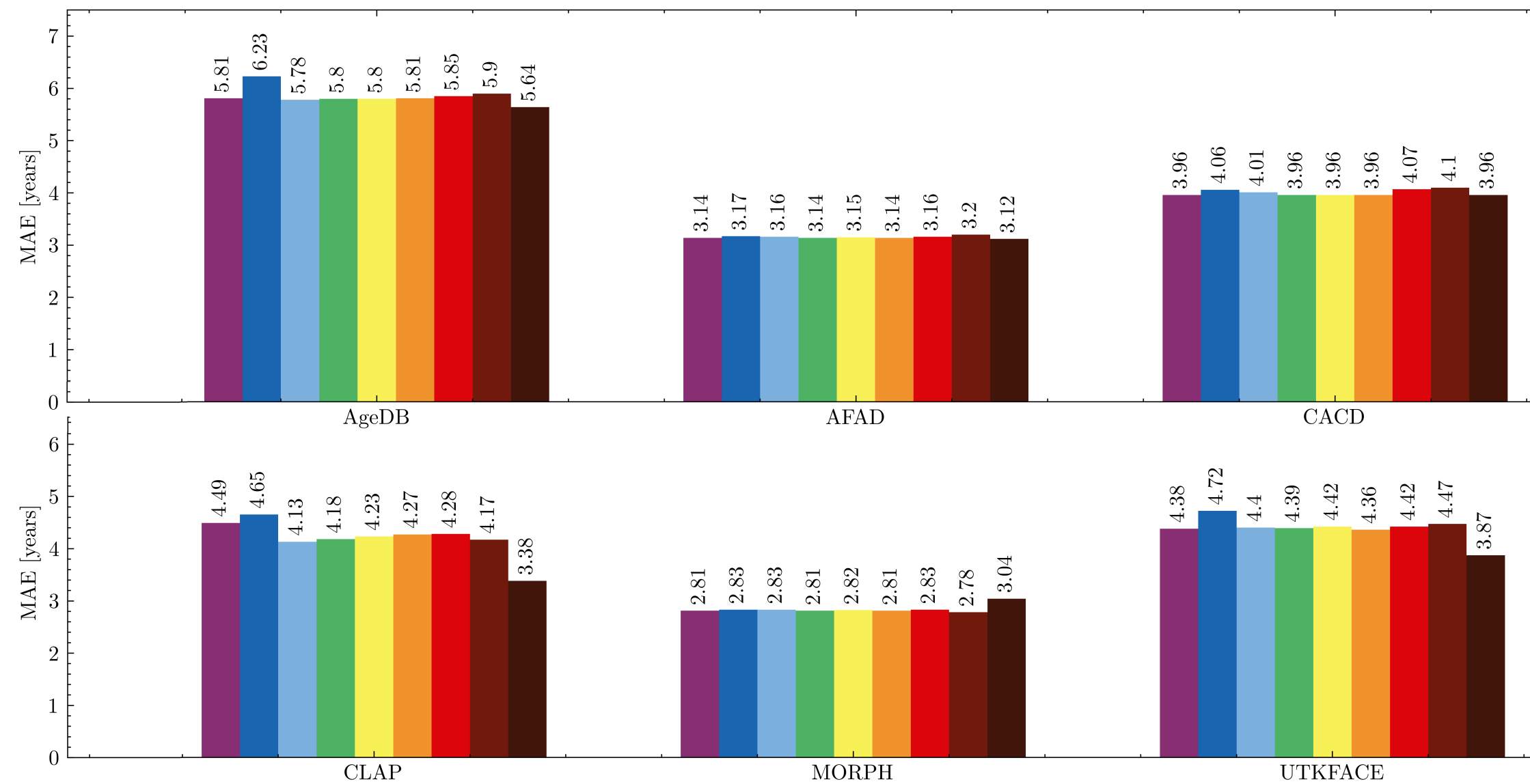
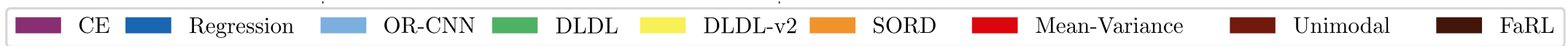


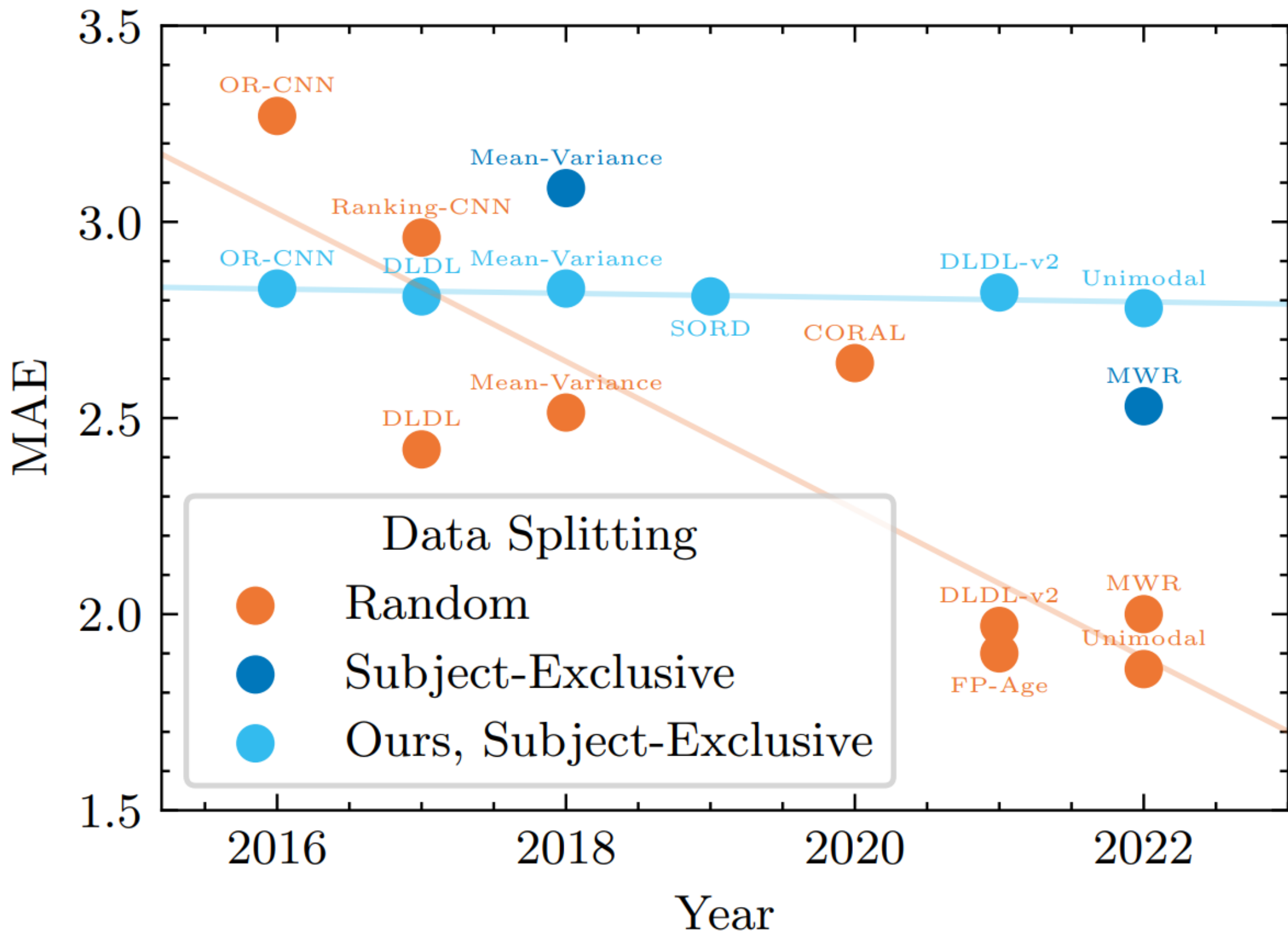
21+



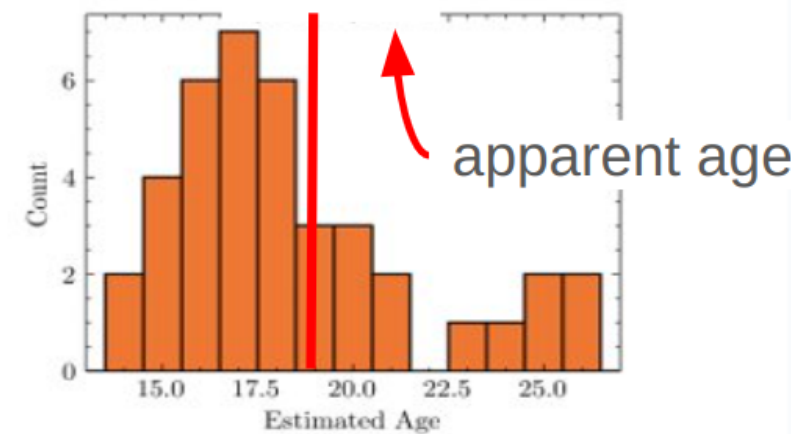
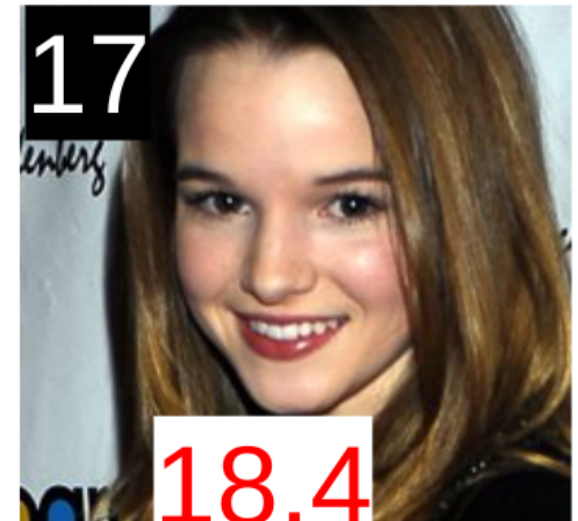
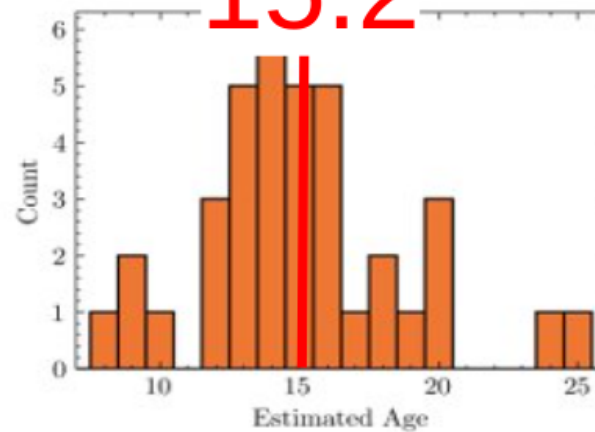
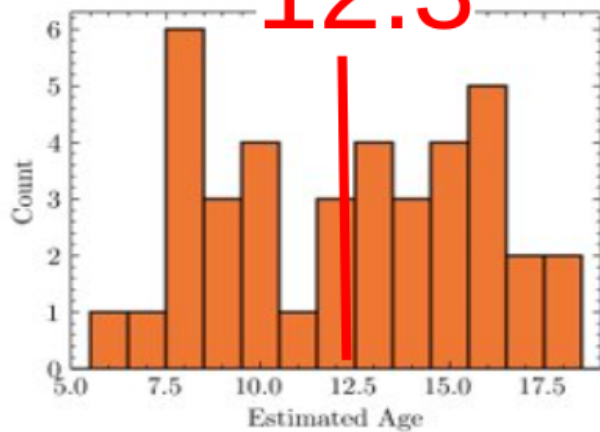
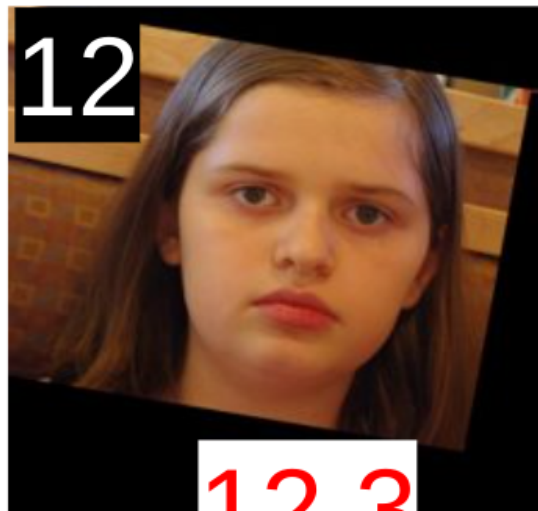


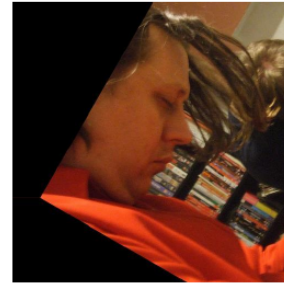
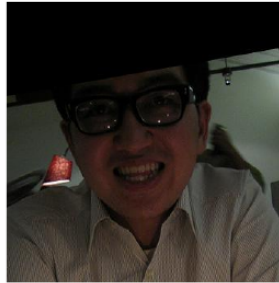






- **Task 1 (main):** Real age estimation
- **Task 2:** Apparent age estimation





	Apparent Age (= avg of 30+ human age estimates)		Real Age	
	[0 – 18] Years	[0 – 90] Years	[0 – 18] Years	[0 – 90] Years
CTU-FaceAge	1.8	3.6	2.7	4.9
Human	1.4	3.9	3.2	6.0
Previous SOTA	NA	4.1	NA	5.2



Happy



Neutral



Angry



Disgust



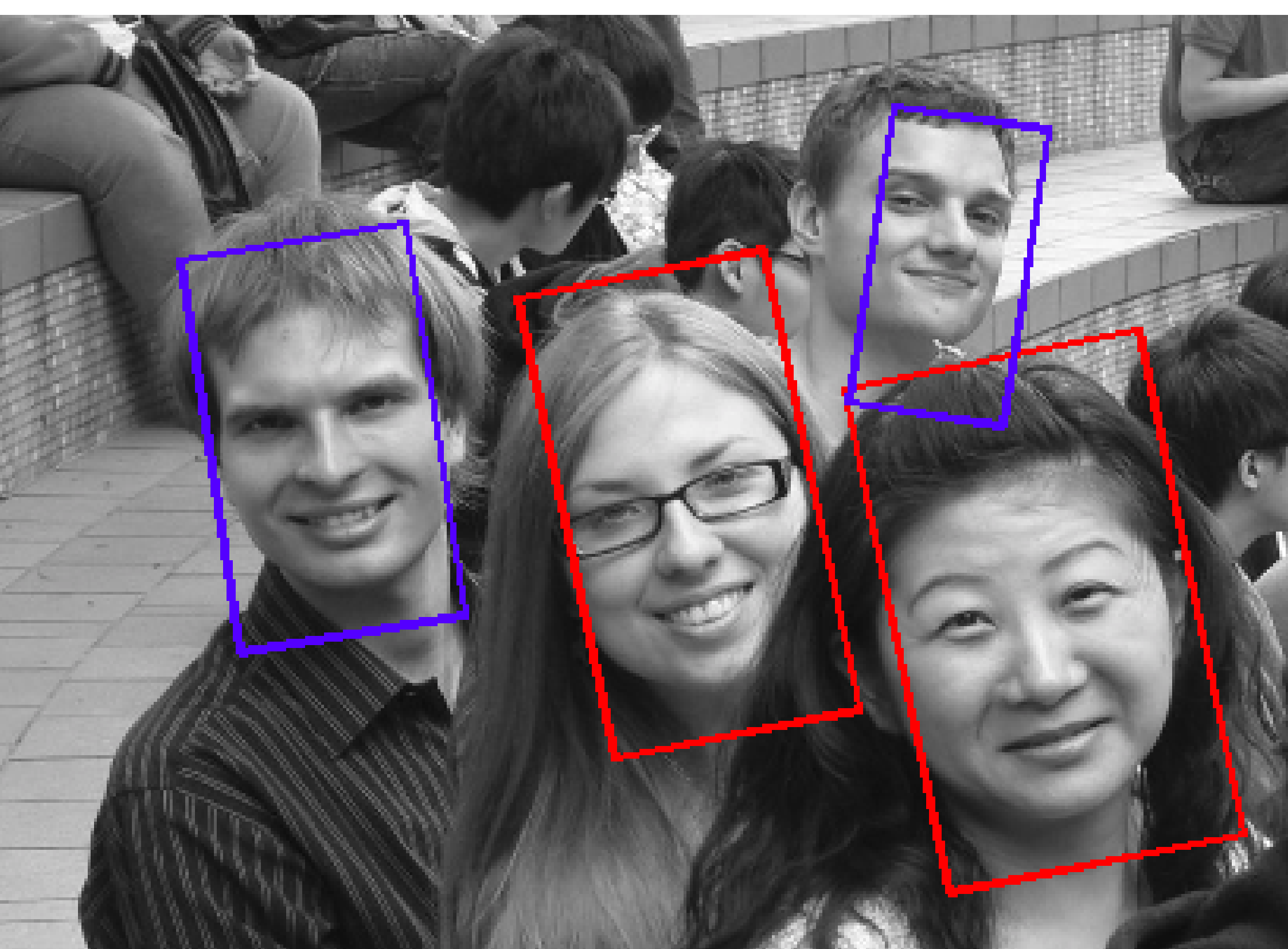
Fear



Sad



Surprise



Eyeglasses



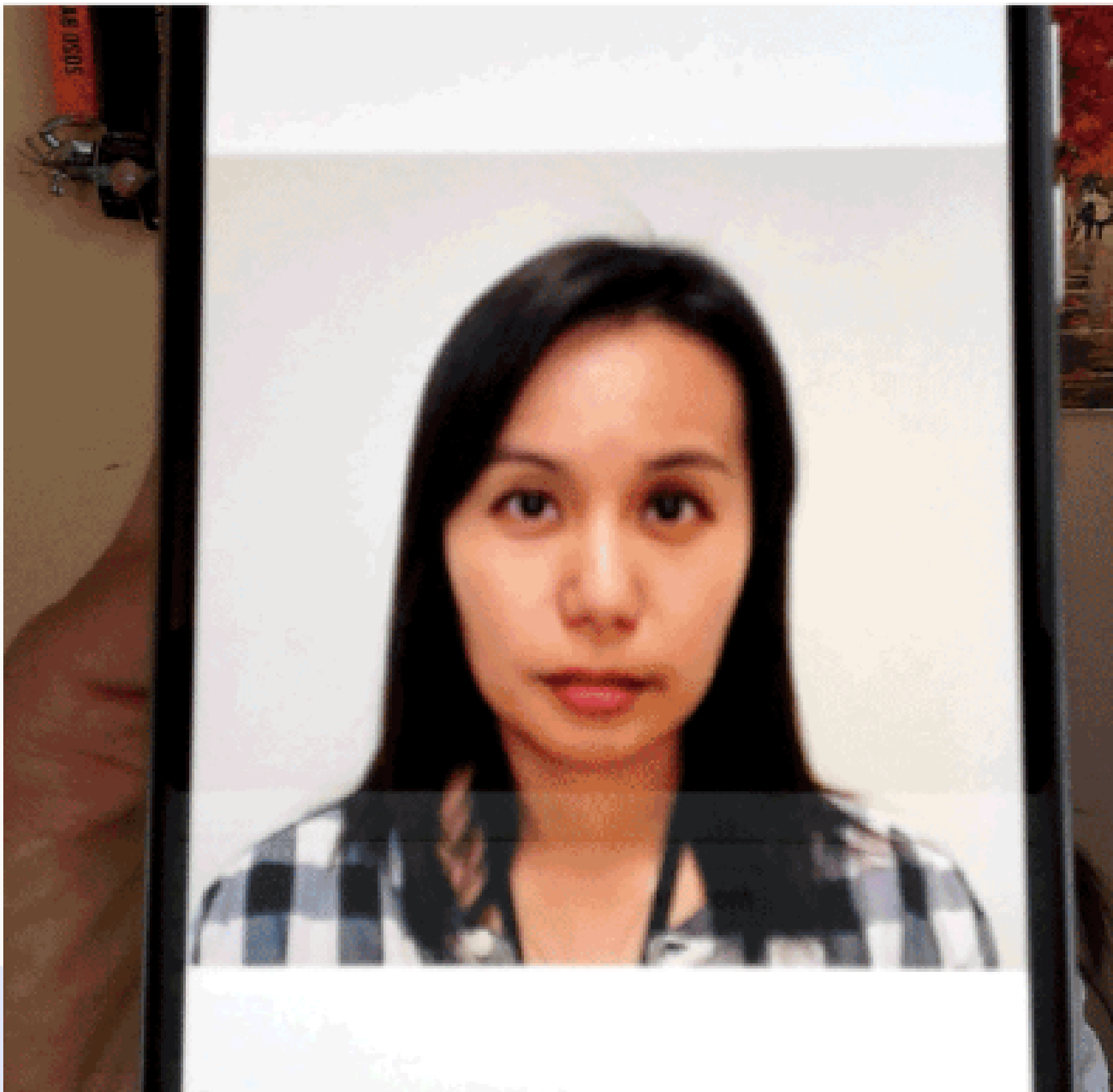
Wearing
Hat



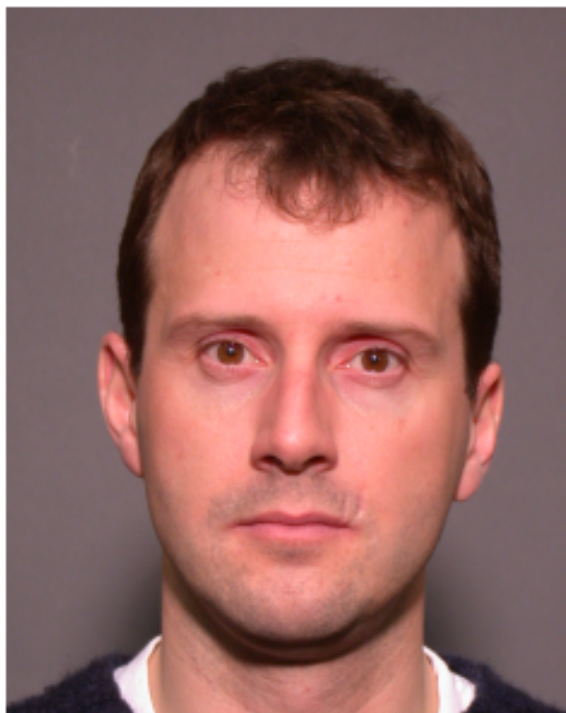
Pointy
Nose











Q = 95



Q = 85



Q = 62



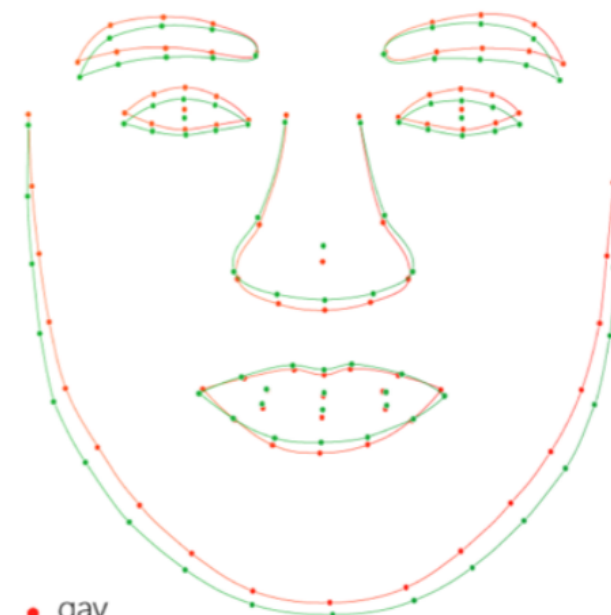
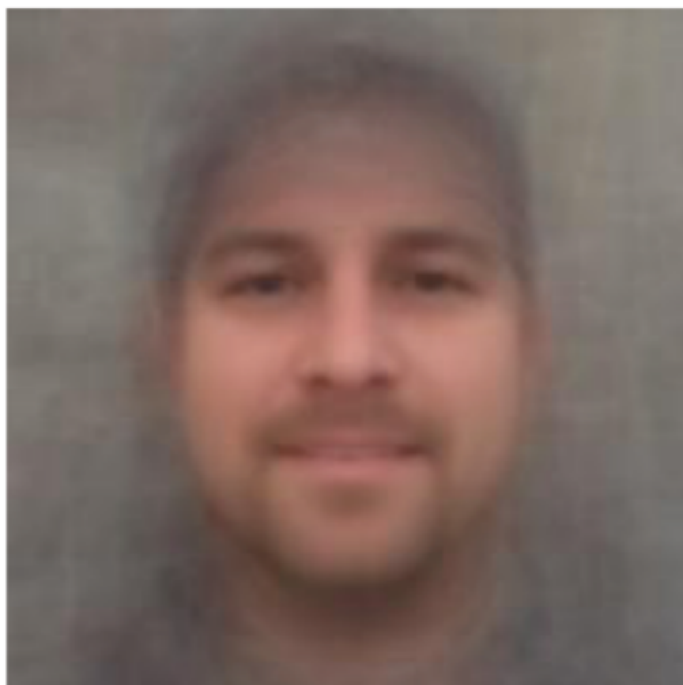
Q = 42

Composite heterosexual faces

Composite gay faces

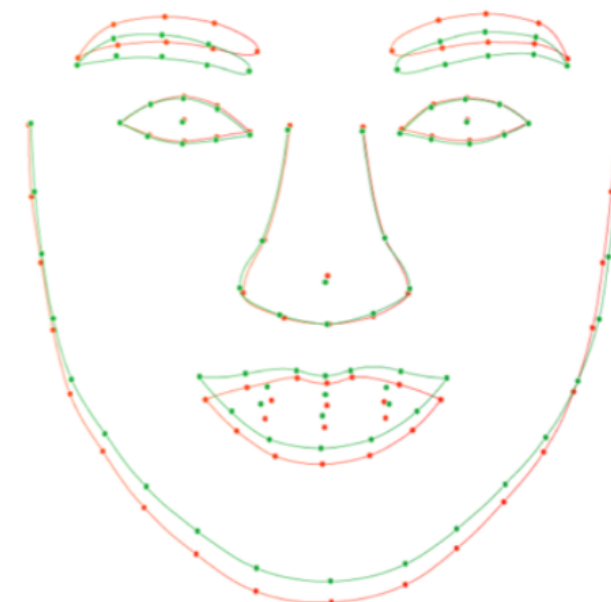
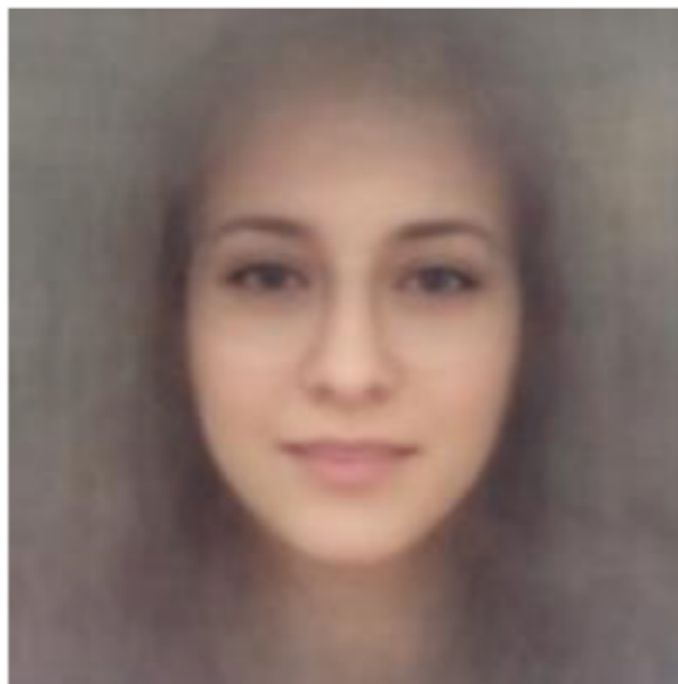
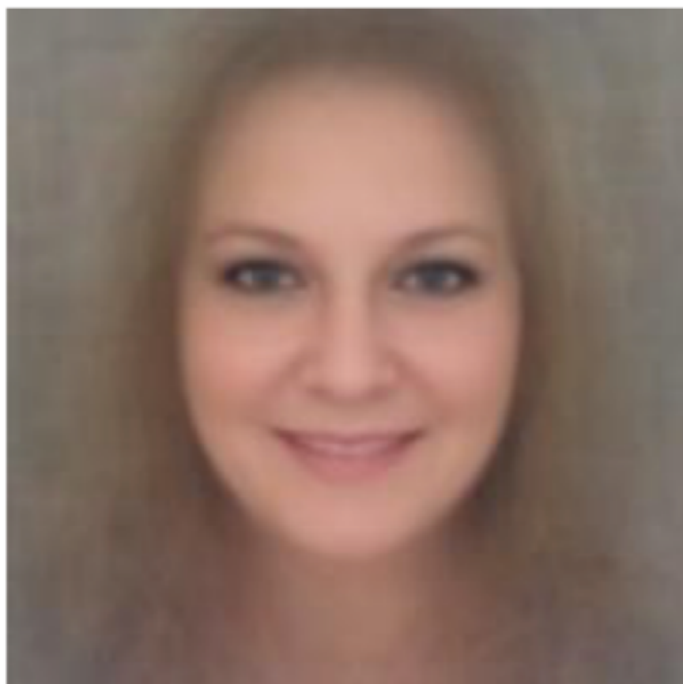
Average facial landmarks

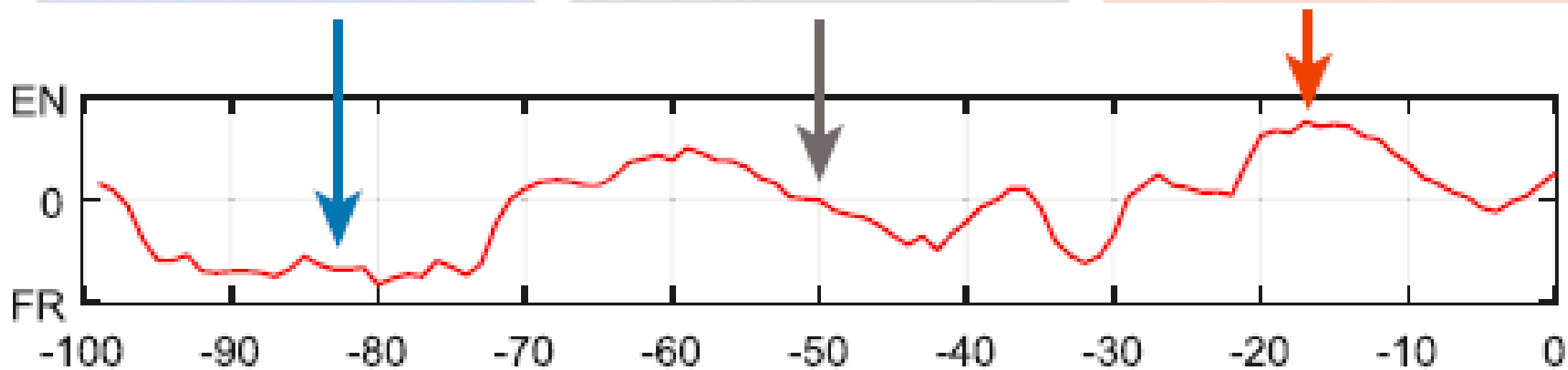
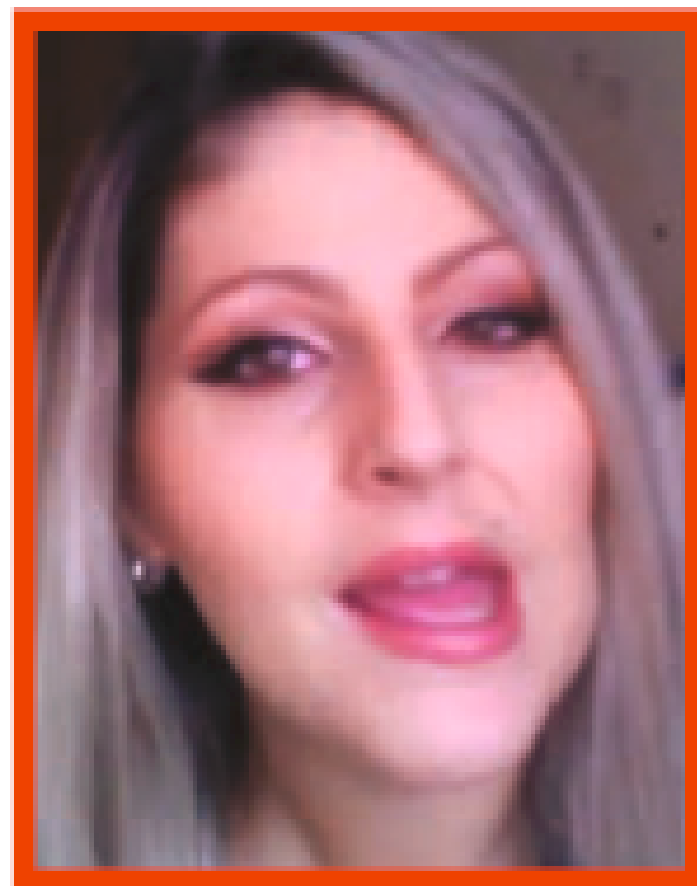
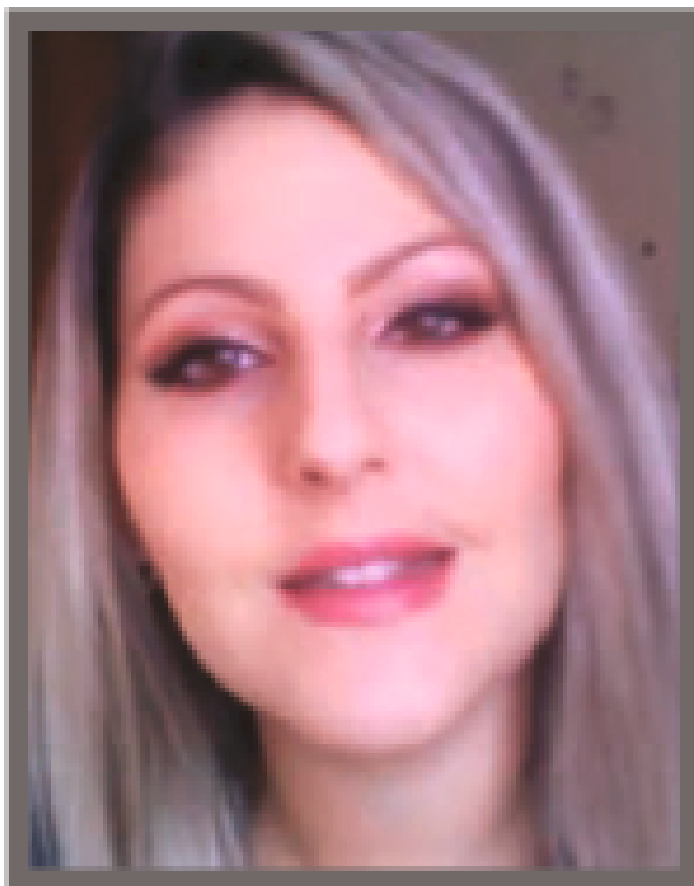
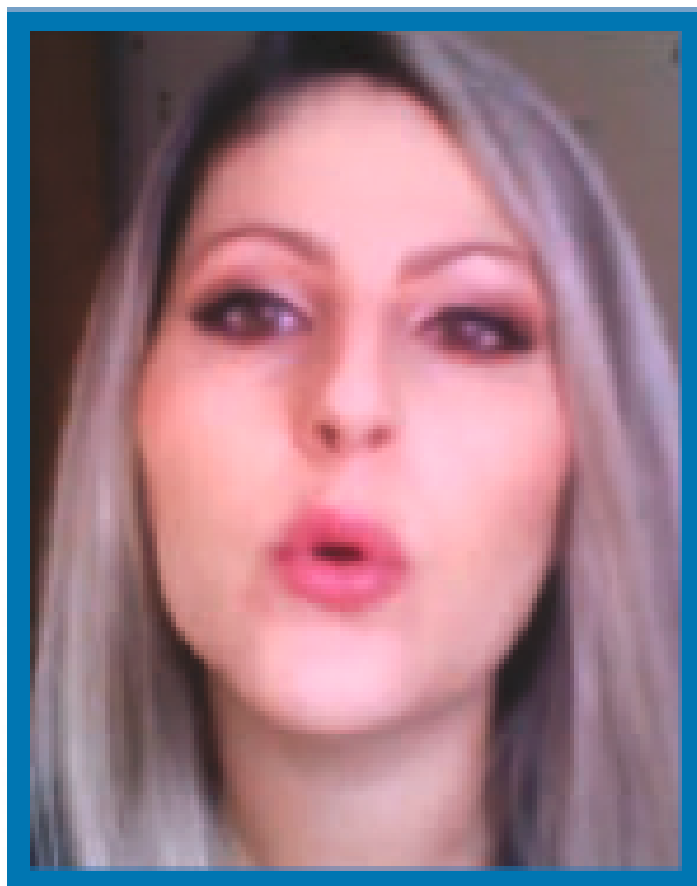
Male

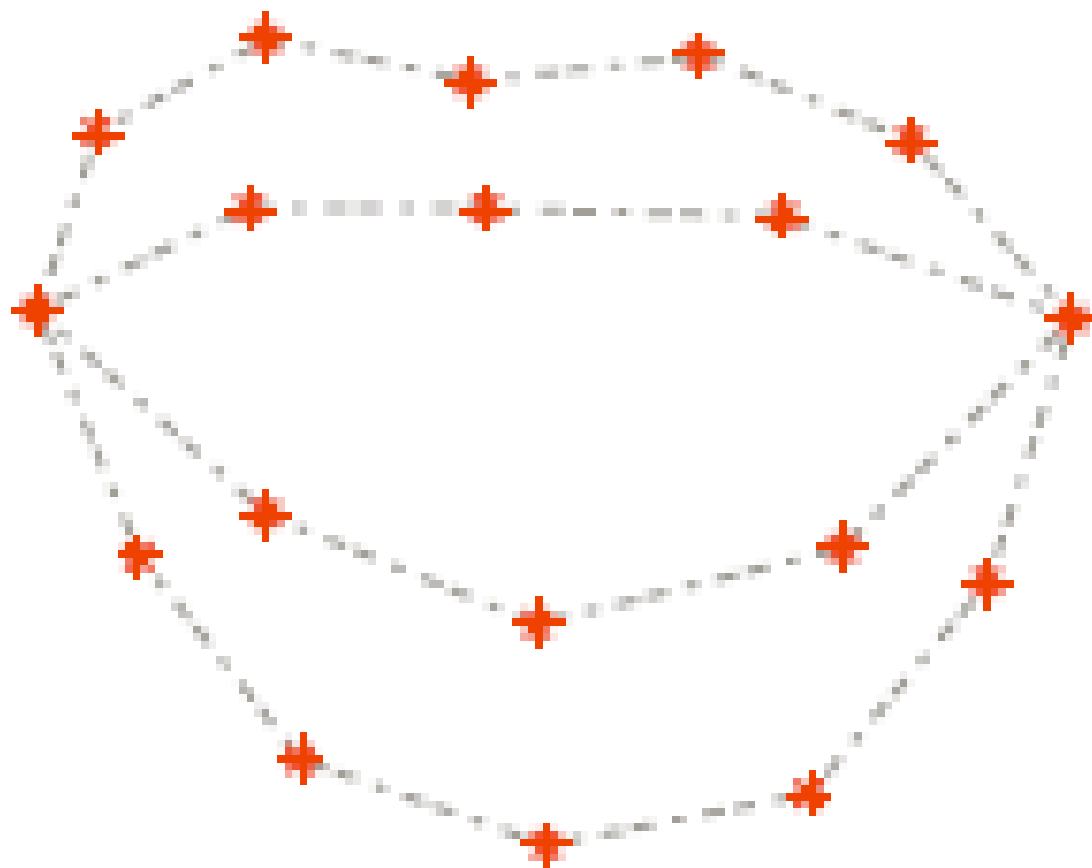
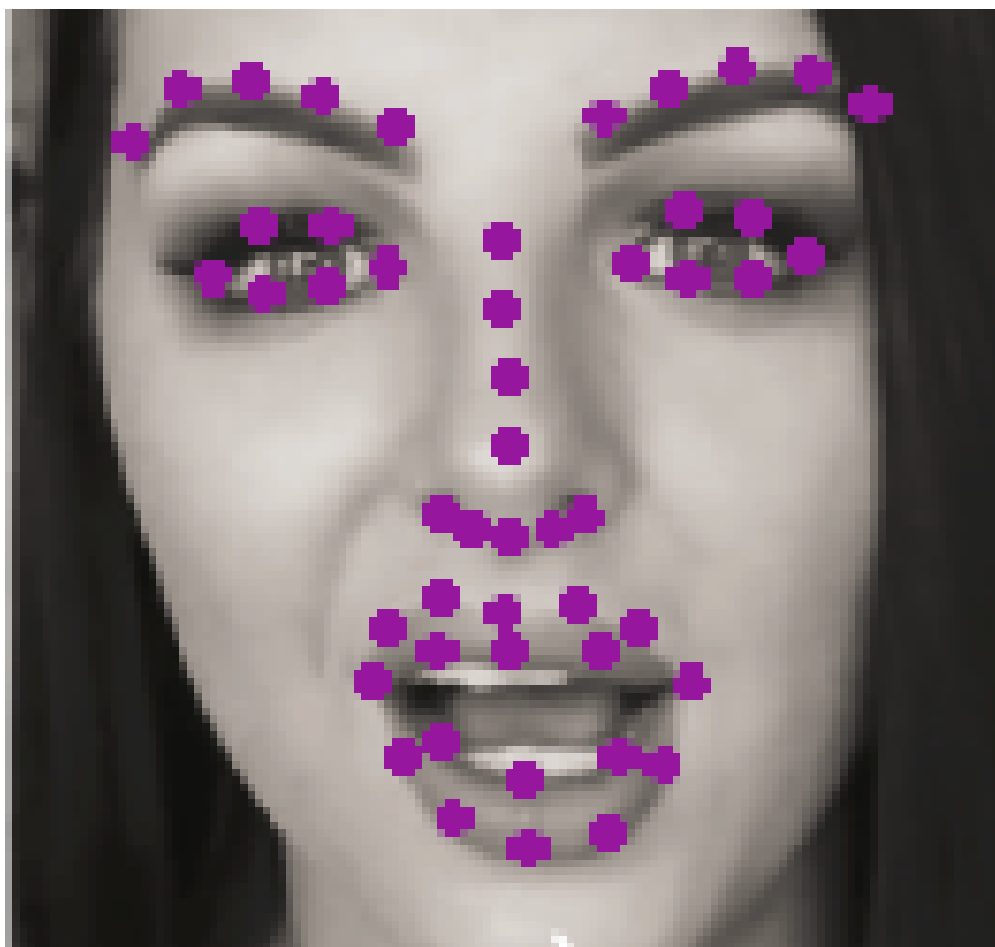


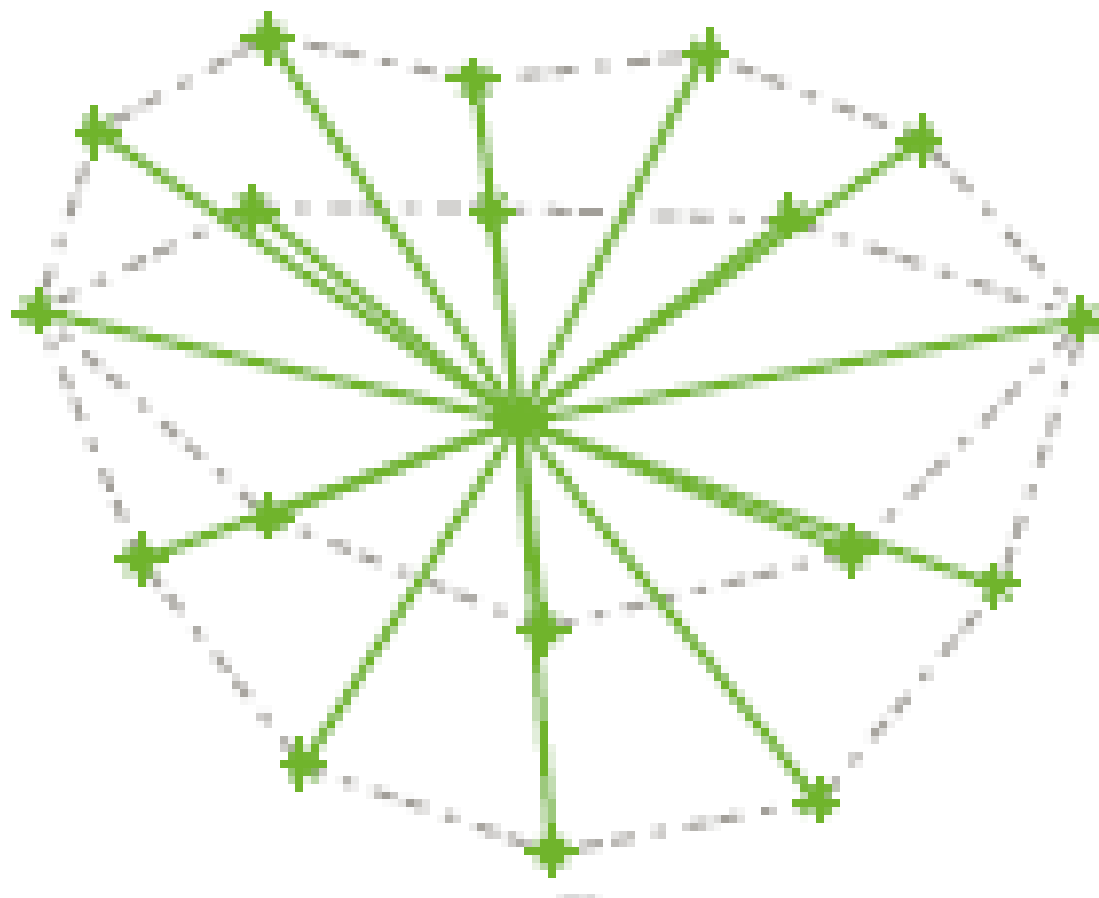
● gay
● straight

Female

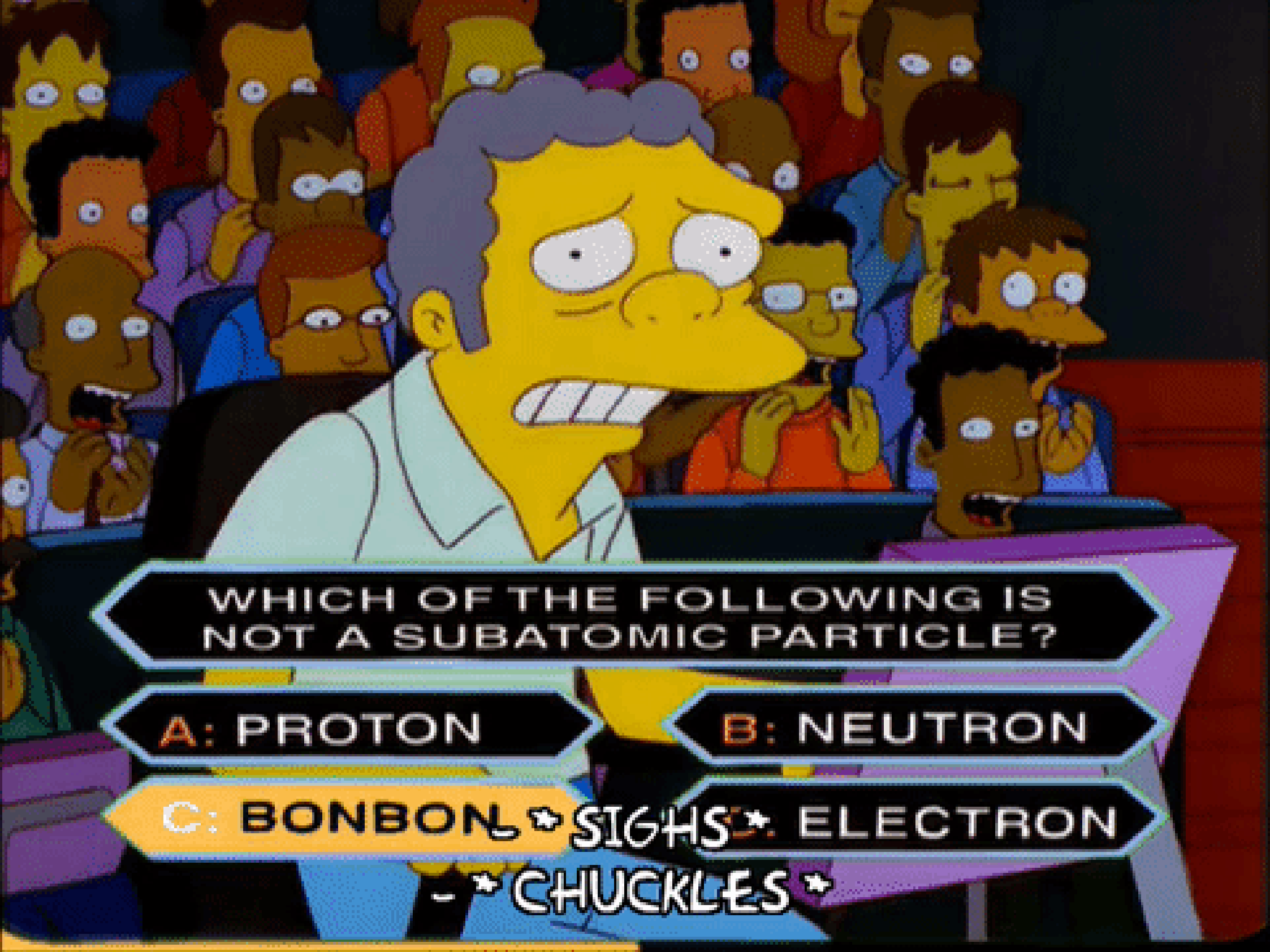












WHICH OF THE FOLLOWING IS
NOT A SUBATOMIC PARTICLE?

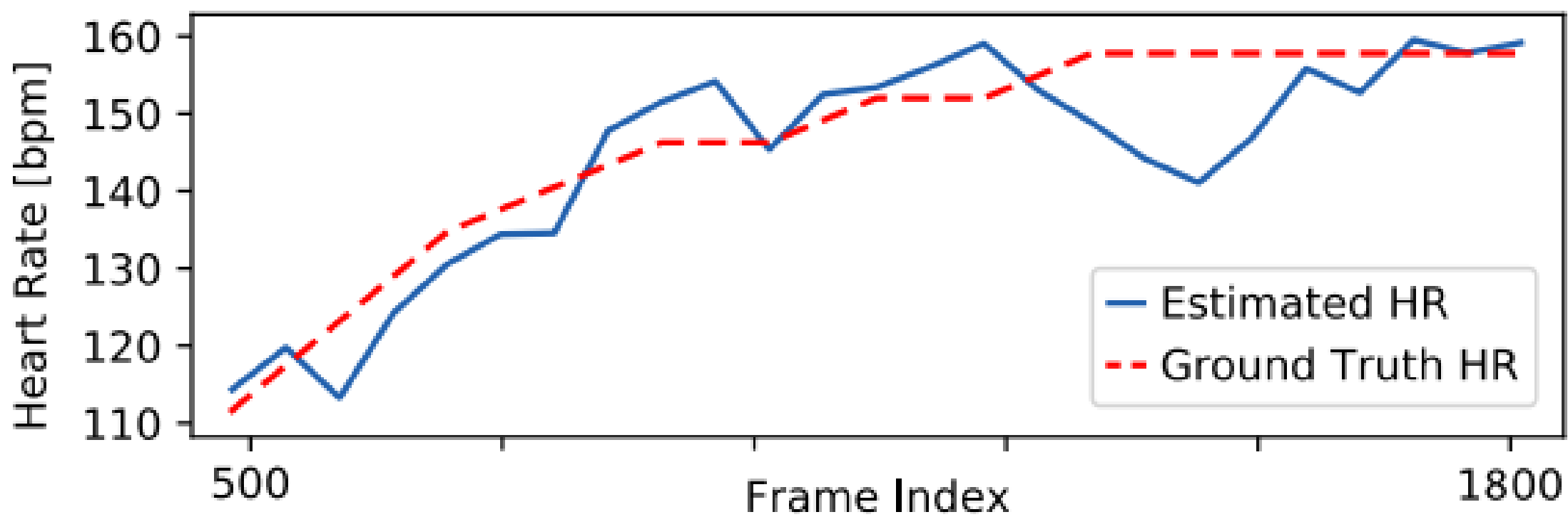
A: PROTON

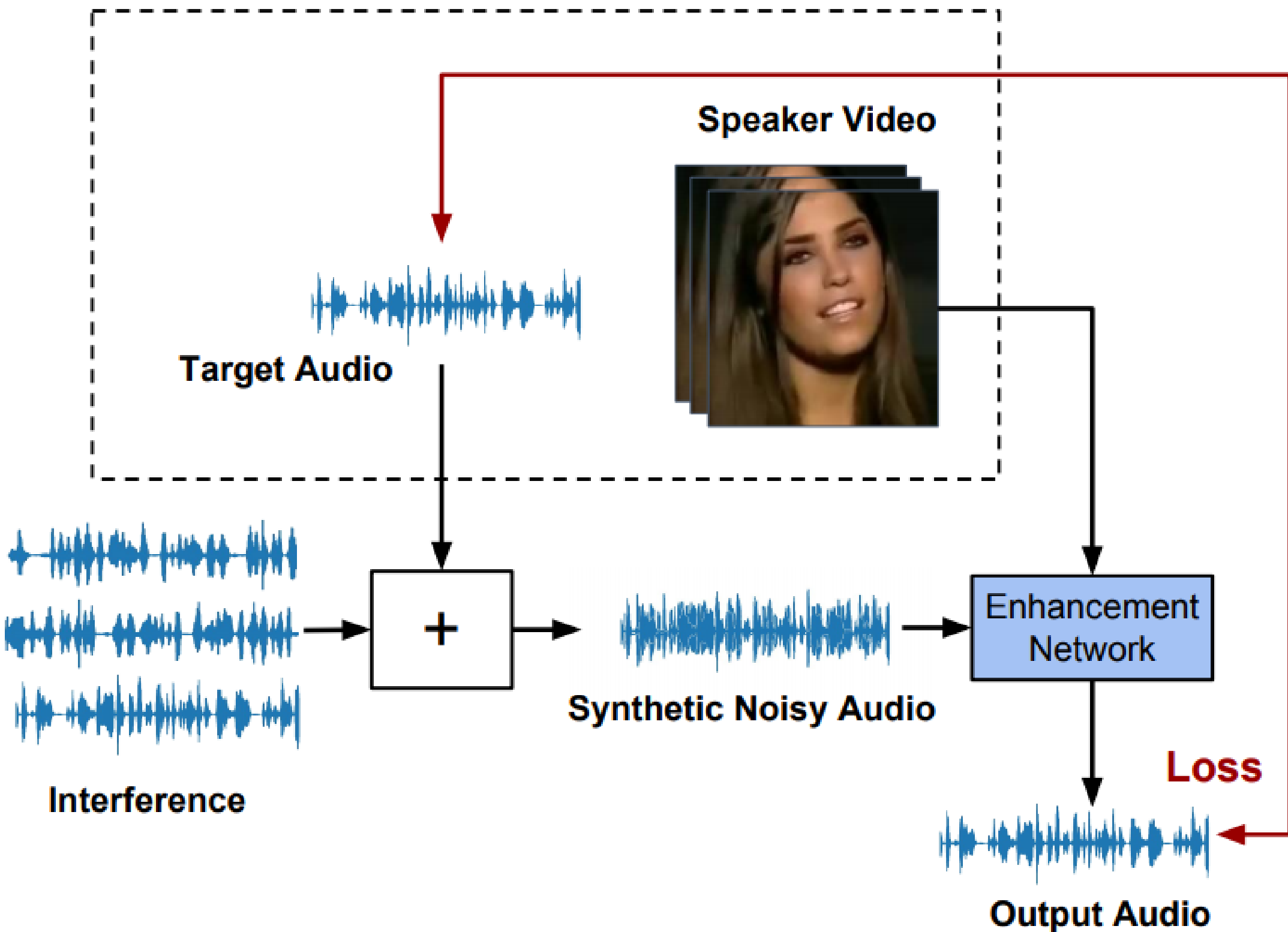
B: NEUTRON

C: BONBON *SIGHS* ELECTRON

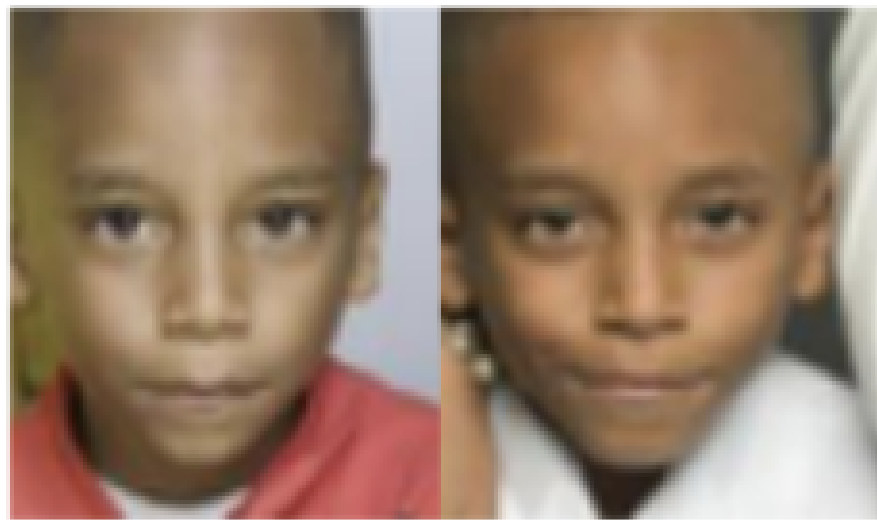
- *CHUCKLES*

		COHFACE	ECG-Fitness	MAHNOB	PURE	PURE MPEG-4 Visual
MAE [bmp]	2SR	20.98	43.66	13.84	2.44	5.78 ①
	CHROM	7.80 ①	21.37	13.49	2.07	6.29
	LiCVPR	19.98	63.25	7.41	28.22	28.39
	SAMC*	—	—	—	—	—
	HR-CNN	8.10	14.48 ①	7.25 ①	1.84 ①	8.72

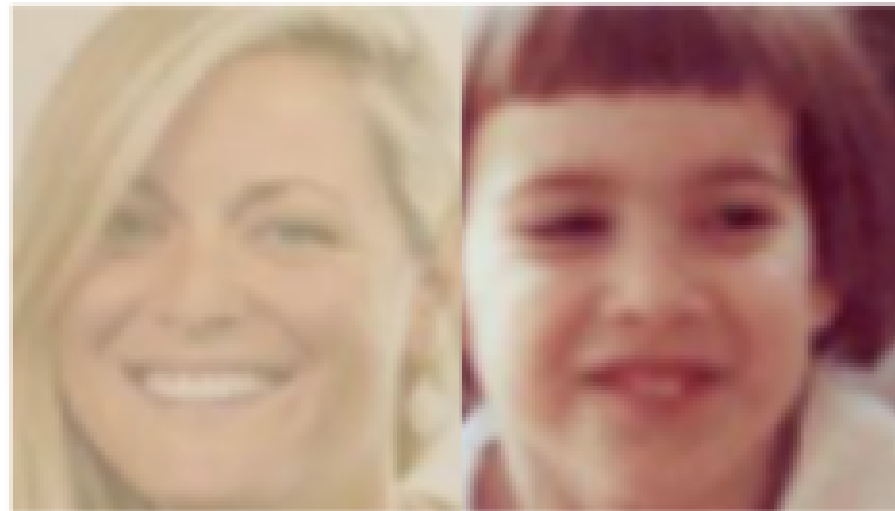




BB



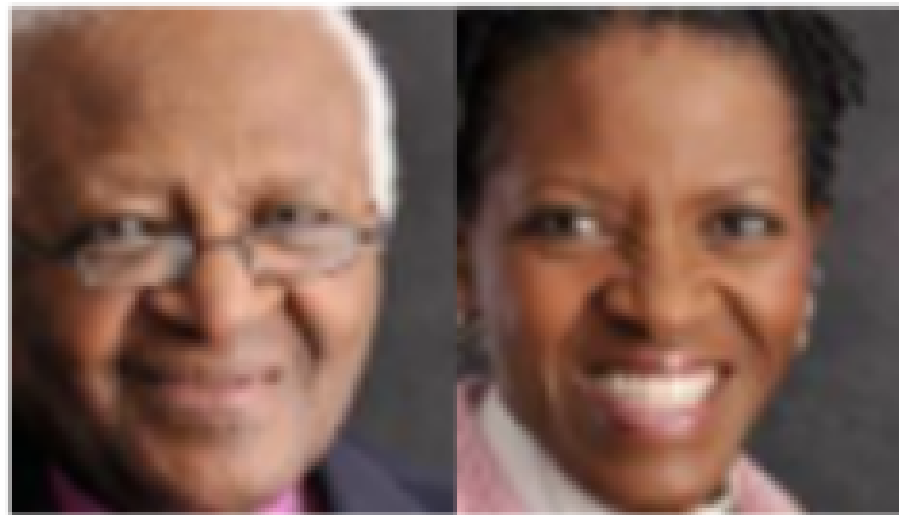
SS



SIBS



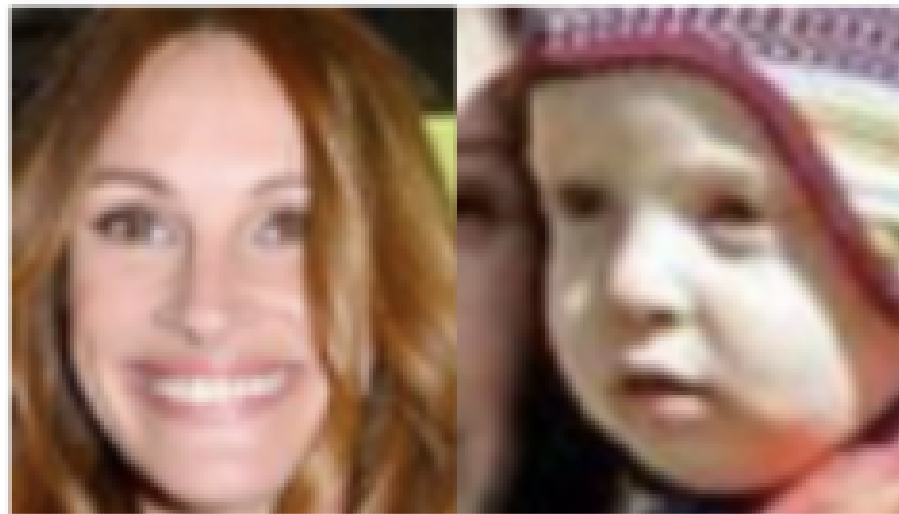
FD

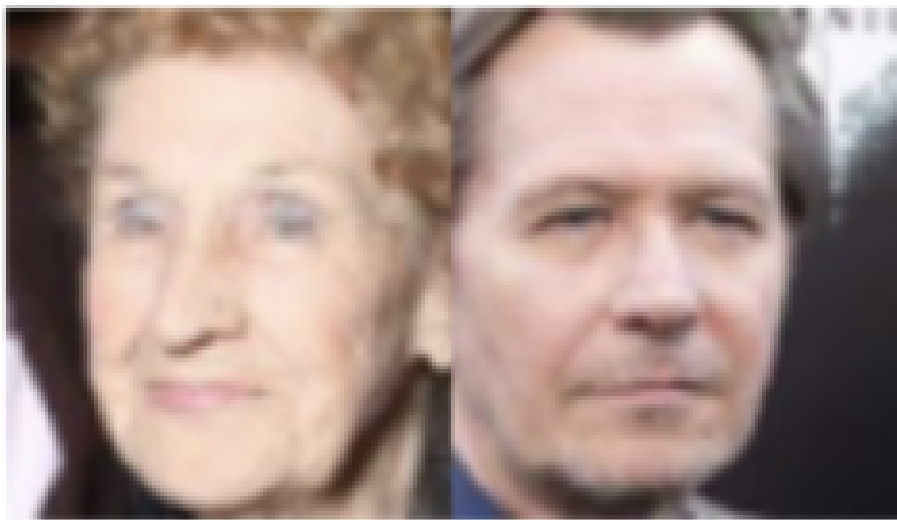


FS



MD





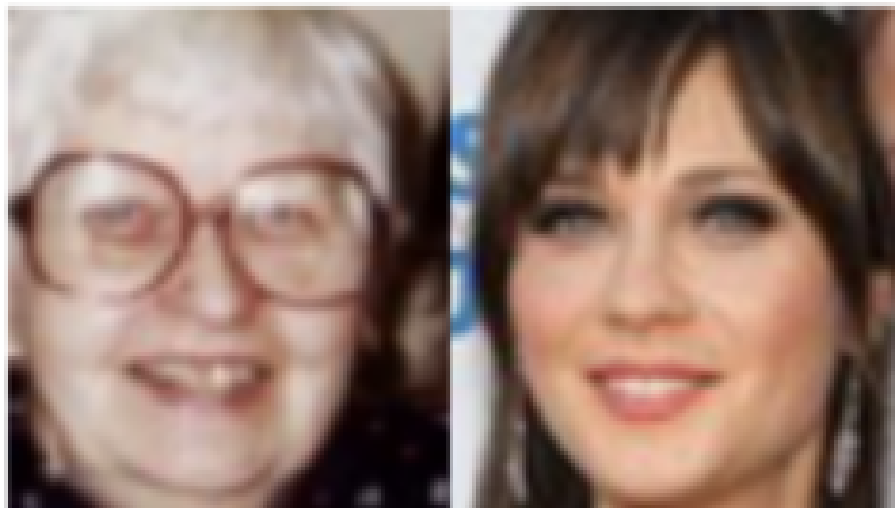
MS



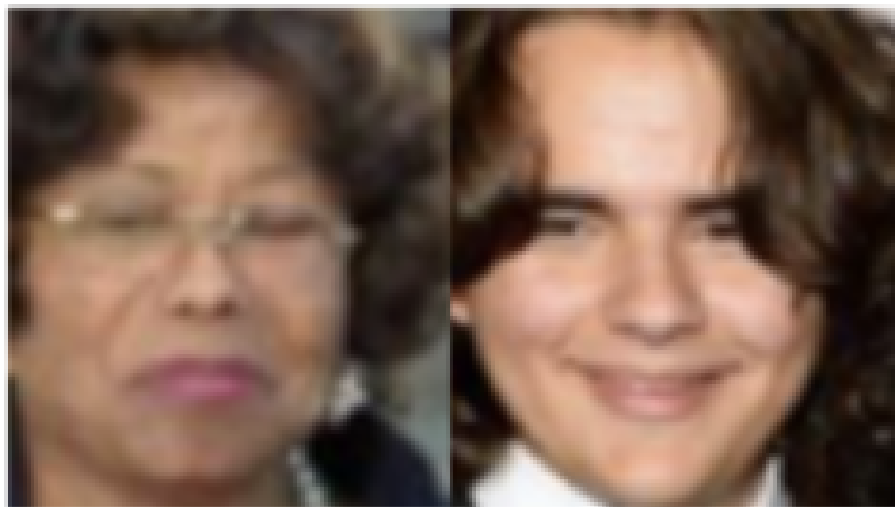
GFGD



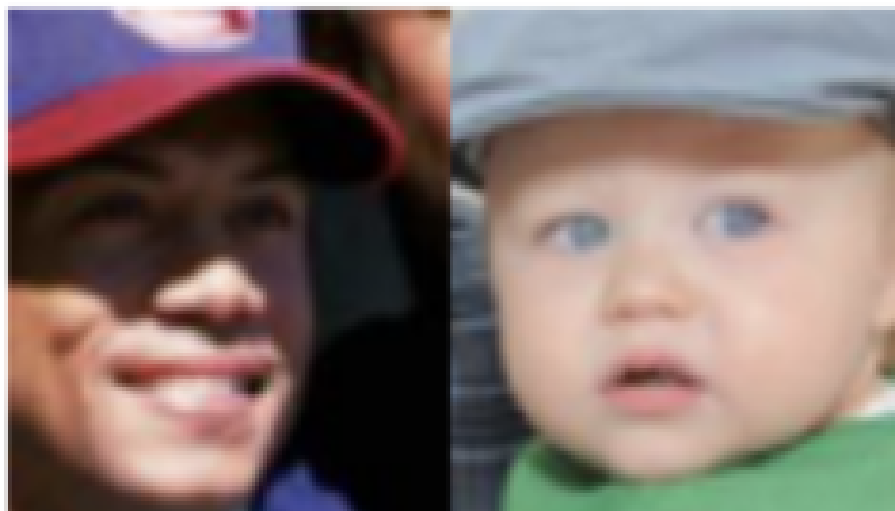
GFGS



GMGD



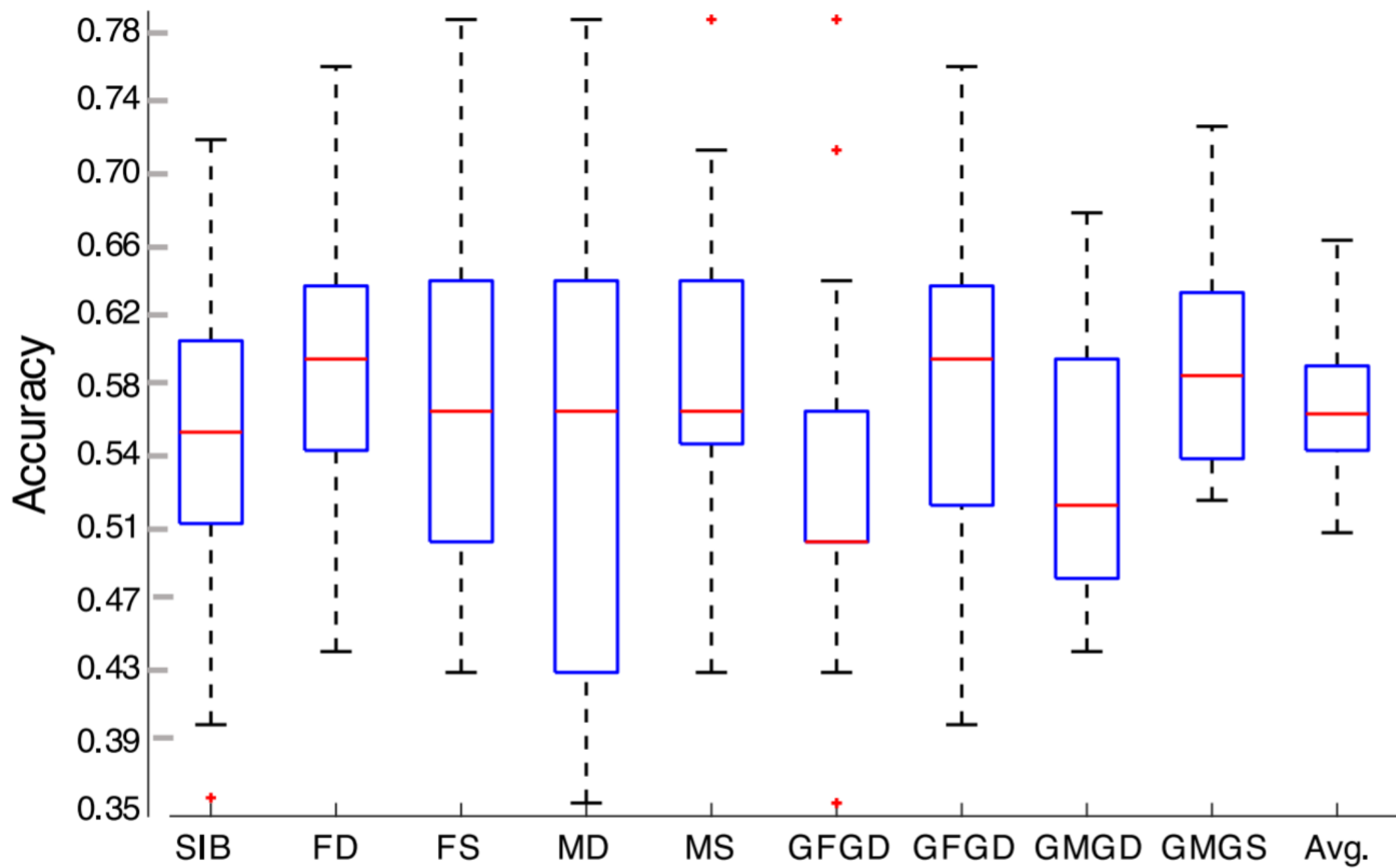
GMGS



NON-
KIN

Averaged Verification Accuracies (%) for 5-Fold Experiment on FIW with No Family Overlap Between Folds

Method	siblings			parent-child				grandparent-grandchild				Acc. $\pm$ Std.
	B-B	S-S	SIBS	F-D	F-S	M-D	M-S	GF-GD	GF-GS	GM-GD	GM-GS	
LBP [56]	55.52	57.49	55.39	55.05	53.77	55.69	54.65	55.79	55.92	54.00	55.36	55.33 $\pm$ 1.01
SIFT [57]	57.86	59.34	56.91	56.37	56.24	55.05	56.45	57.25	55.35	57.29	56.74	56.80 $\pm$ 1.17
ResNet-22 [58]	65.57	69.65	60.12	59.45	60.27	61.45	59.37	55.37	58.15	59.74	59.70	61.34 $\pm$ 3.81
VGG-Face [36]	69.67	75.35	66.52	64.25	63.85	66.43	62.80	62.06	63.79	57.40	61.64	64.89 $\pm$ 4.68
+ITML [59]	57.15	61.61	56.98	58.07	54.73	57.26	59.09	62.52	59.60	62.08	59.92	59.00 $\pm$ 2.44
+LPP [60]	67.61	66.22	71.01	62.54	61.39	65.04	63.54	63.50	59.96	60.00	63.53	64.03 $\pm$ 3.32
+LMNN [61]	67.11	68.33	66.88	65.66	67.08	68.07	66.16	61.90	60.44	63.68	60.15	65.04 $\pm$ 3.00
+GmDAE [62]	68.05	68.55	67.33	66.53	68.30	68.15	66.71	62.10	63.93	63.84	63.10	66.05 $\pm$ 2.36
+DLML [63]	68.03	68.87	67.97	65.96	68.00	68.51	67.21	62.90	63.96	63.11	63.55	66.19 $\pm$ 2.36
+mDML [41]	69.10	70.15	68.11	67.90	66.24	70.39	67.40	65.20	66.78	63.11	63.45	67.07 $\pm$ 2.44
ResNet+CF [64]	69.88	69.54	69.54	68.15	67.73	71.09	68.63	66.37	66.45	64.81	64.39	67.87 $\pm$ 2.15
SphereFace [65]	71.94	77.30	70.23	69.25	68.50	71.81	69.49	66.07	66.36	64.58	65.40	69.18 $\pm$ 3.68





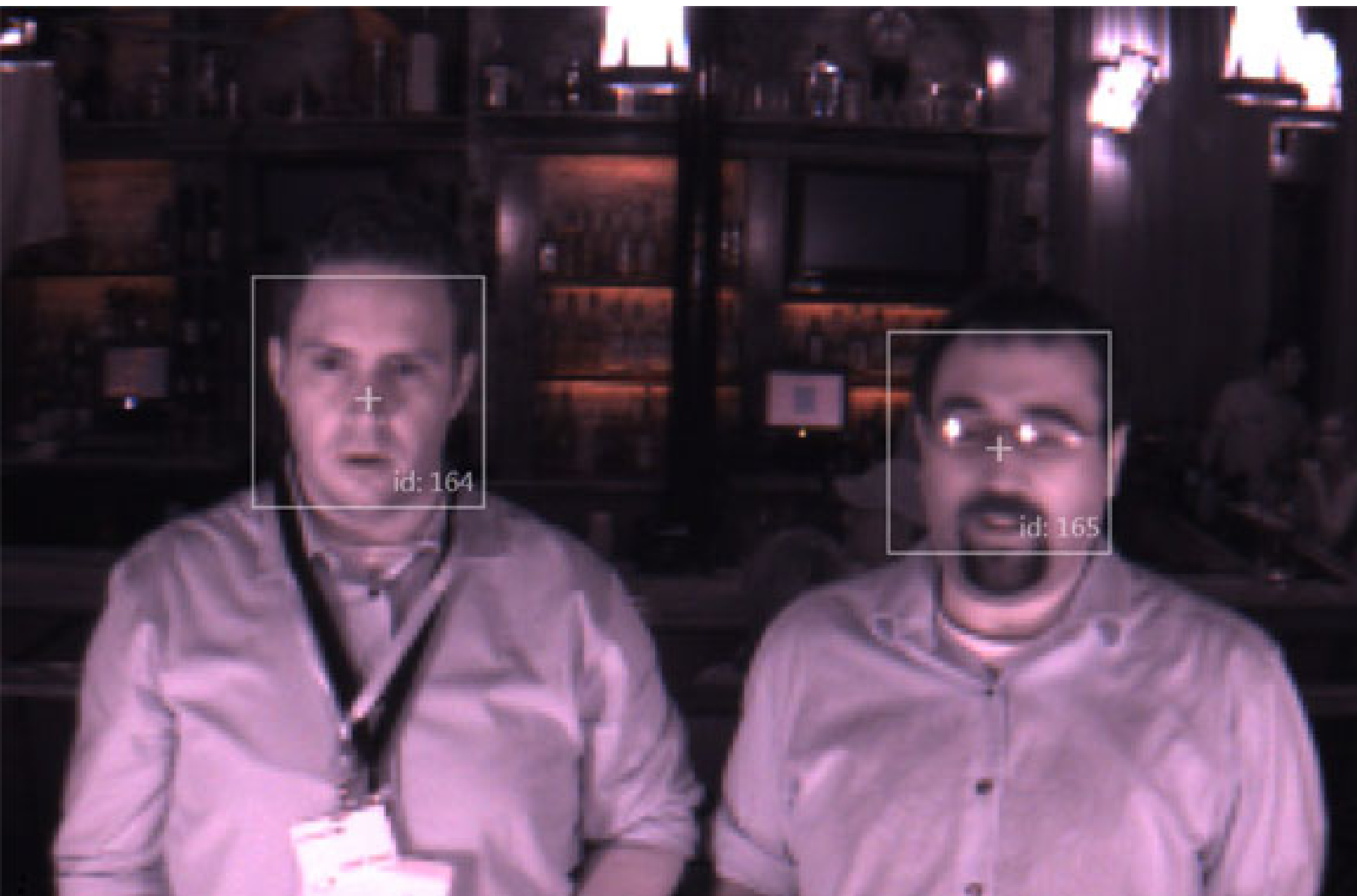










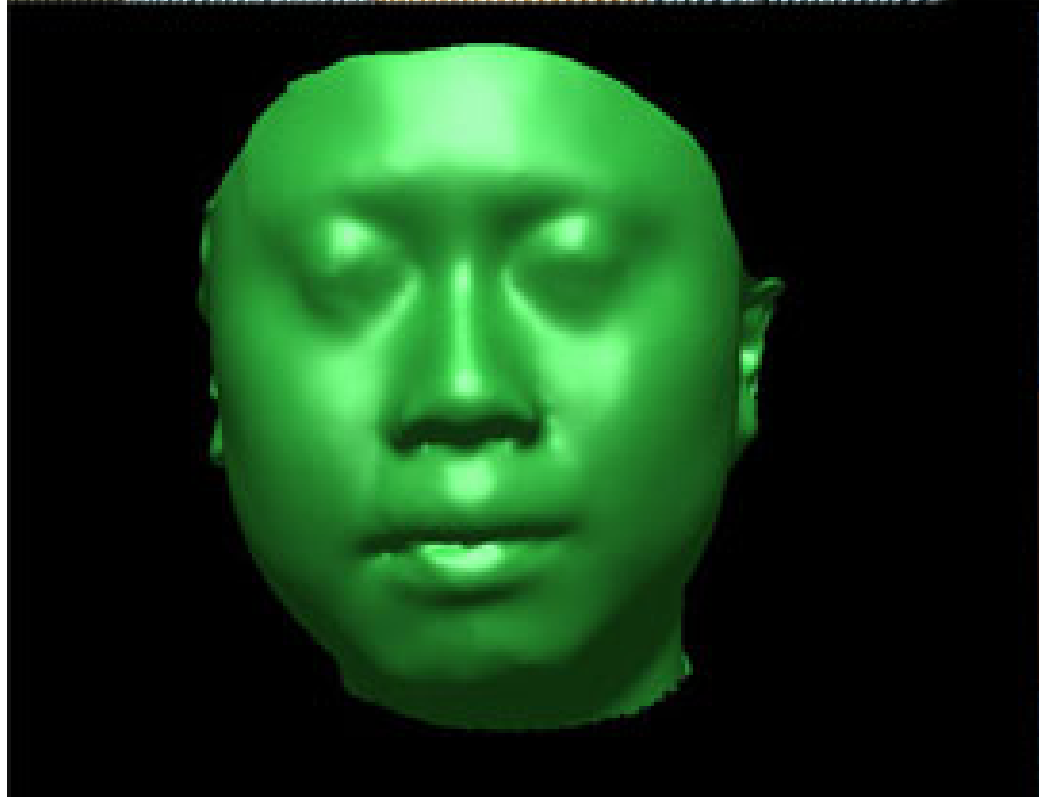


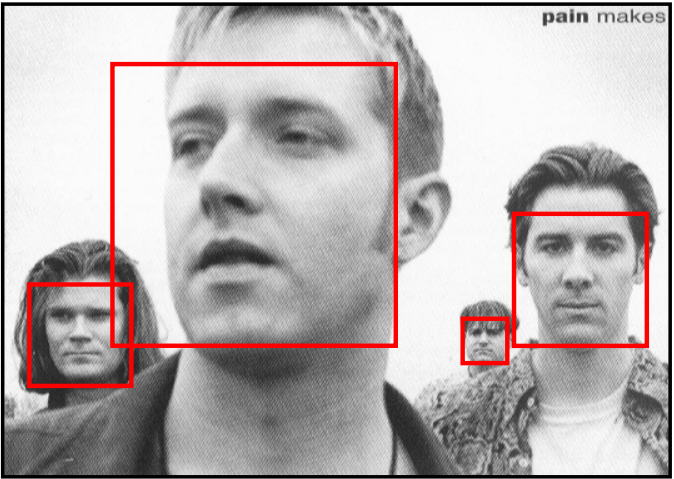
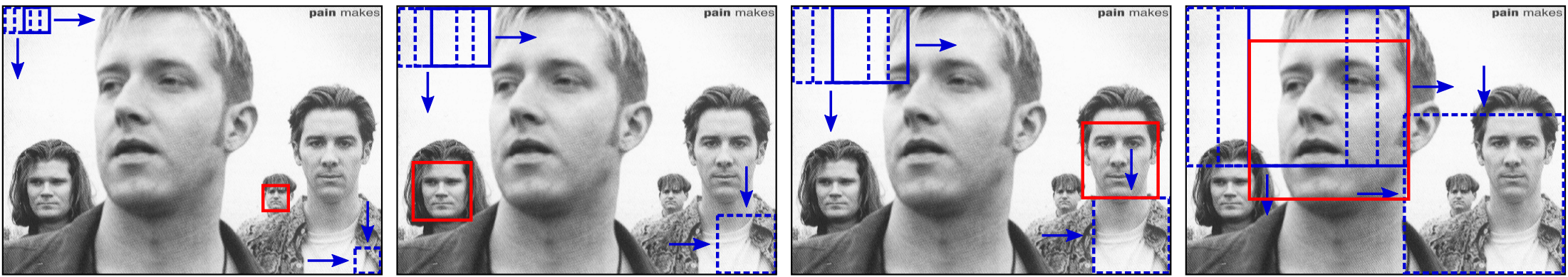
+

id: 164

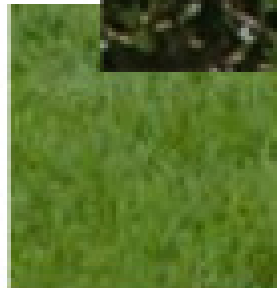
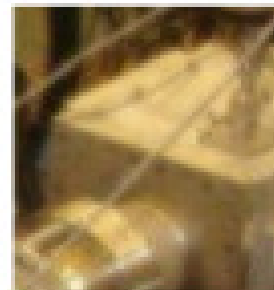
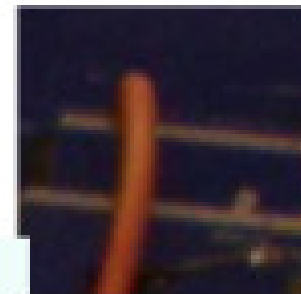
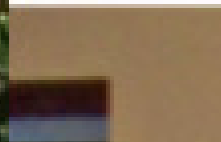
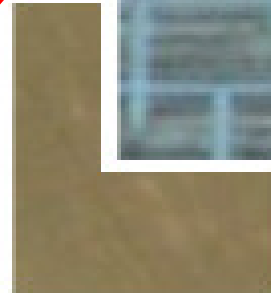
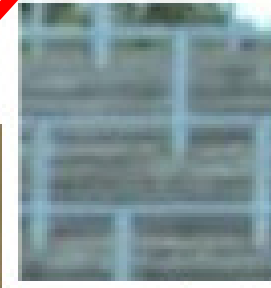
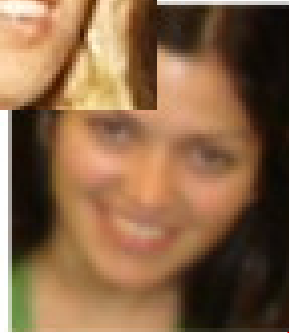
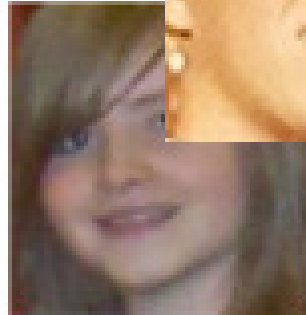
+

id: 165

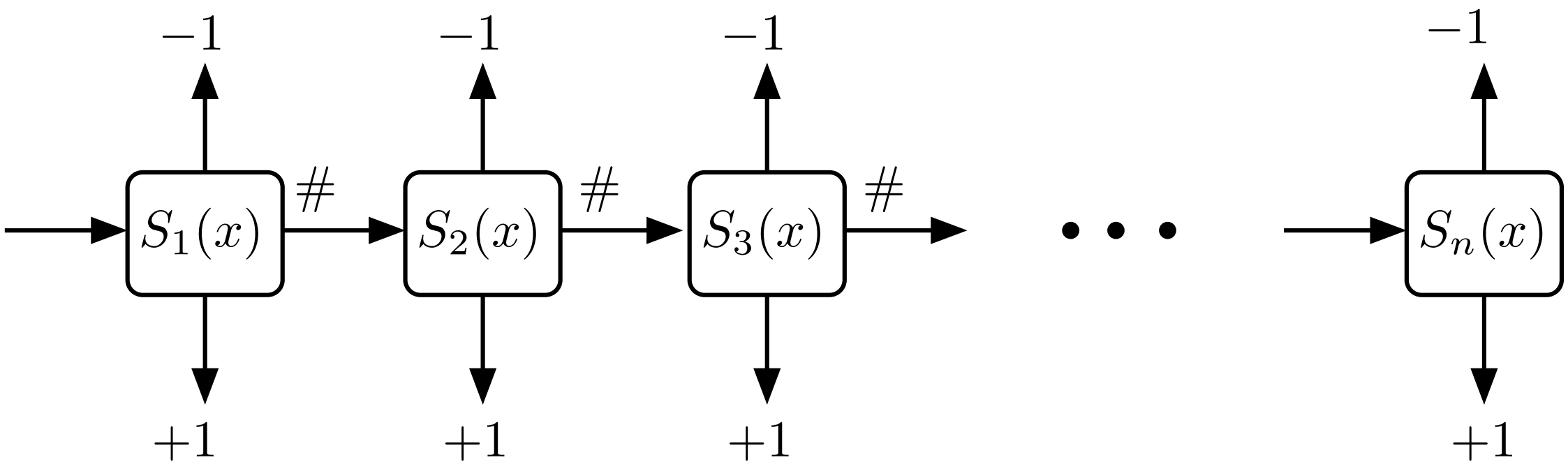




FACE



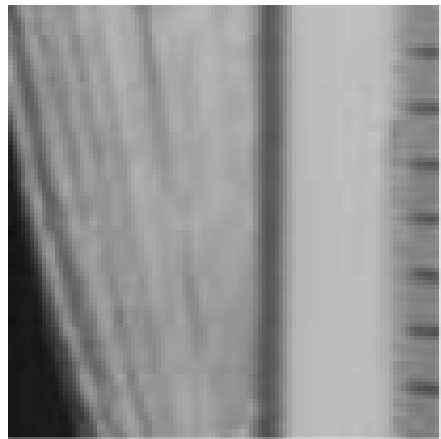
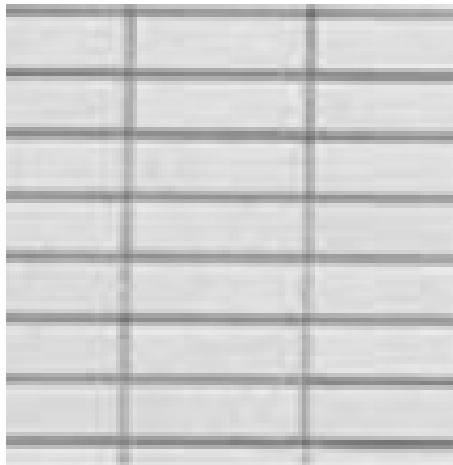
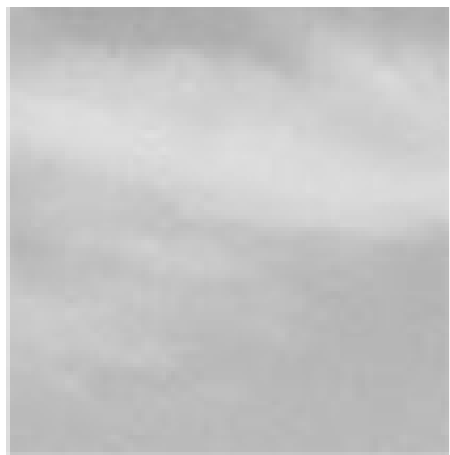
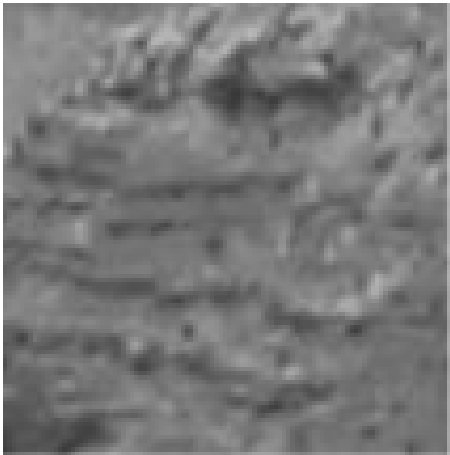
NONFACE

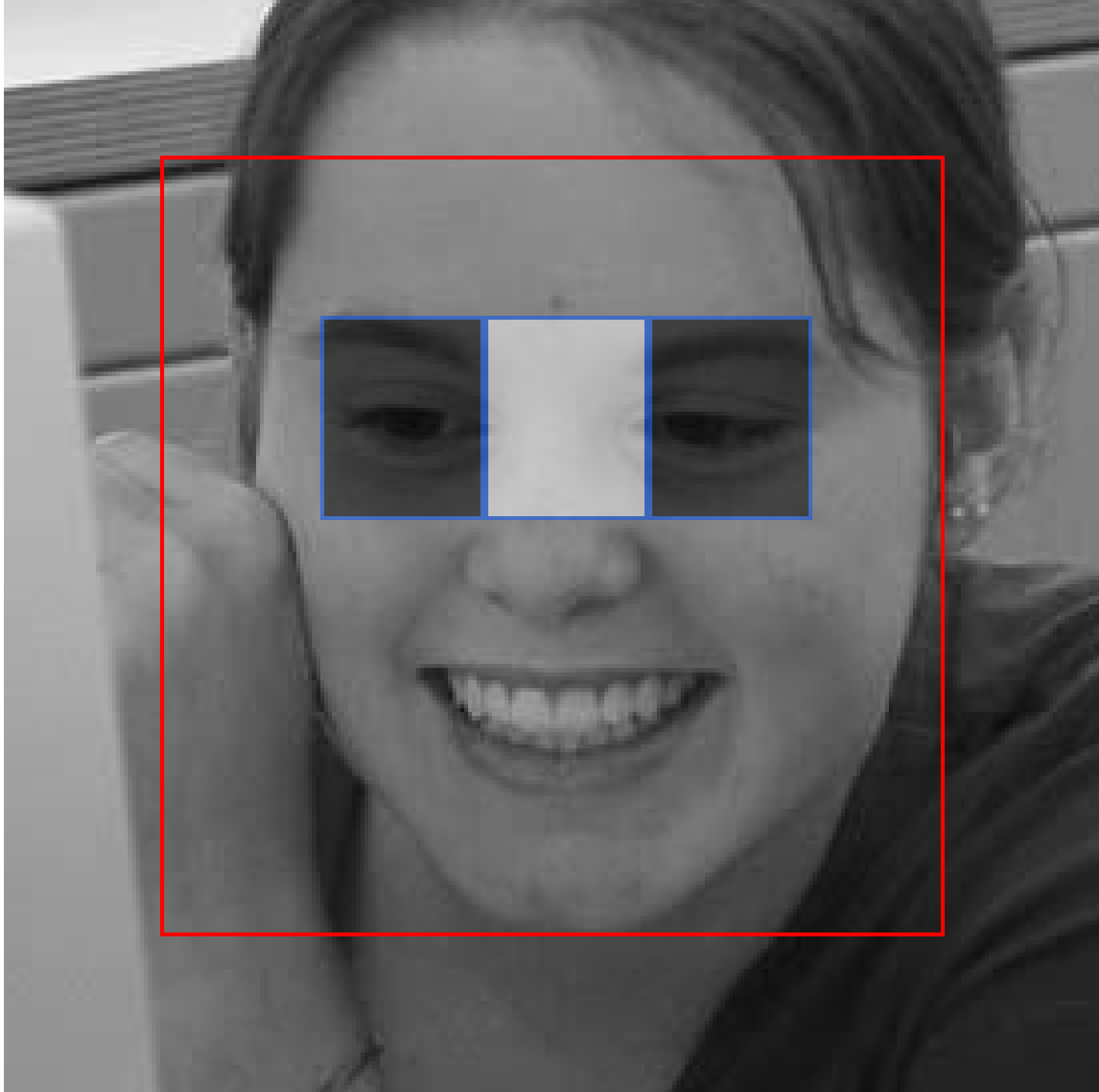


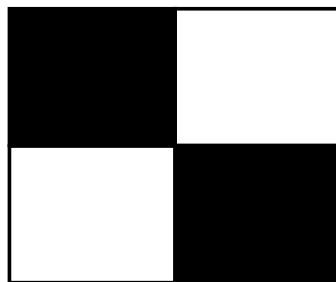
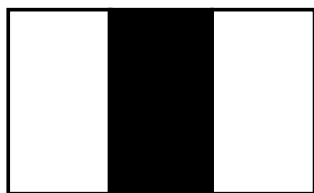
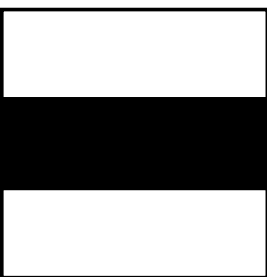
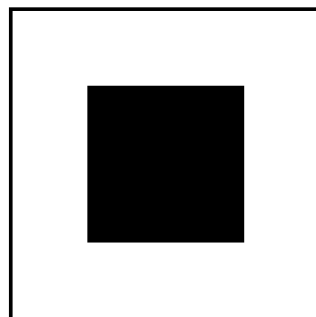
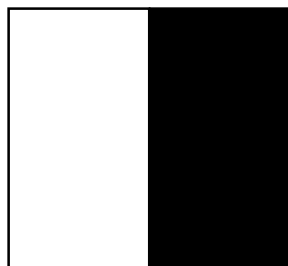


...





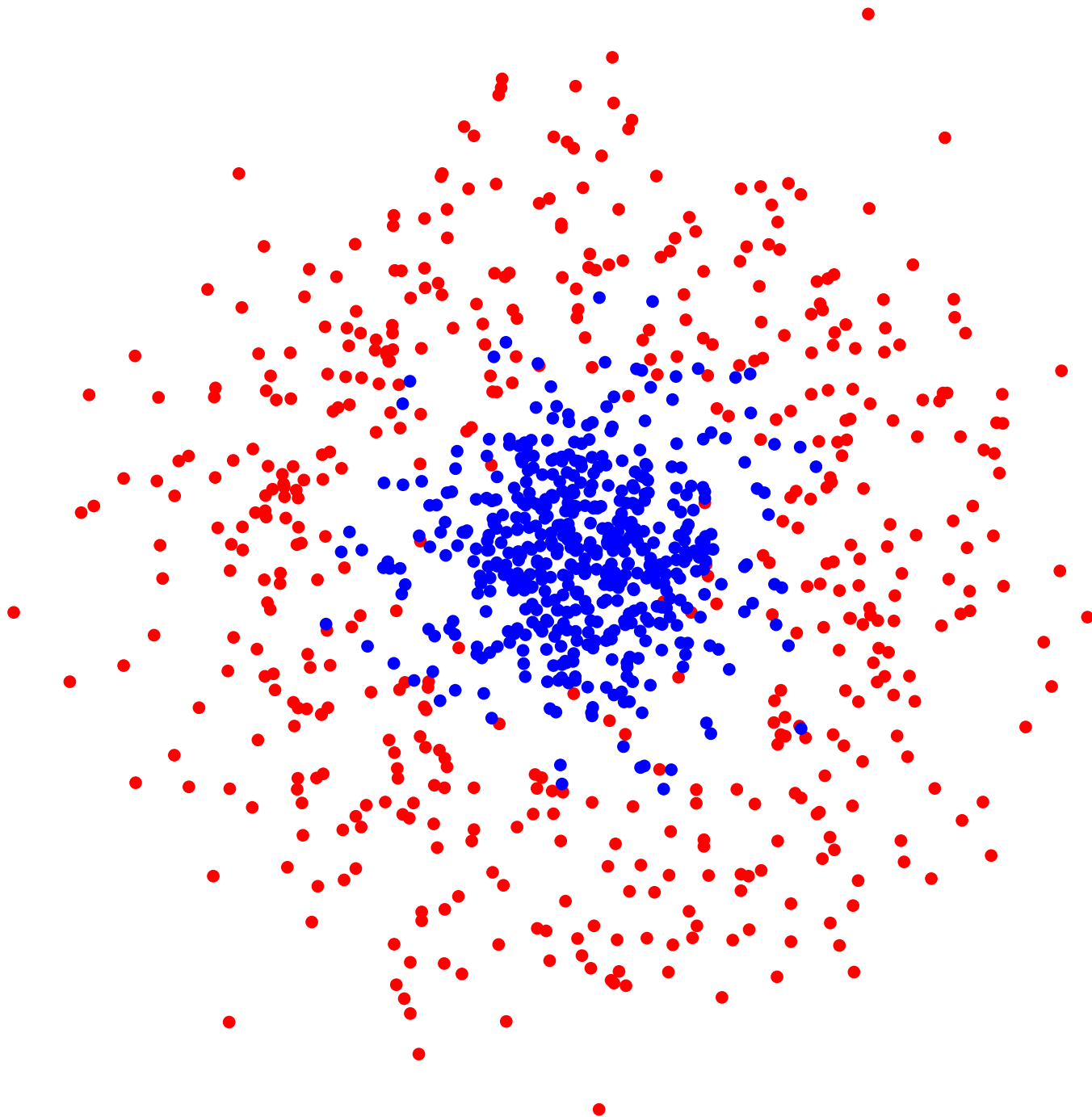


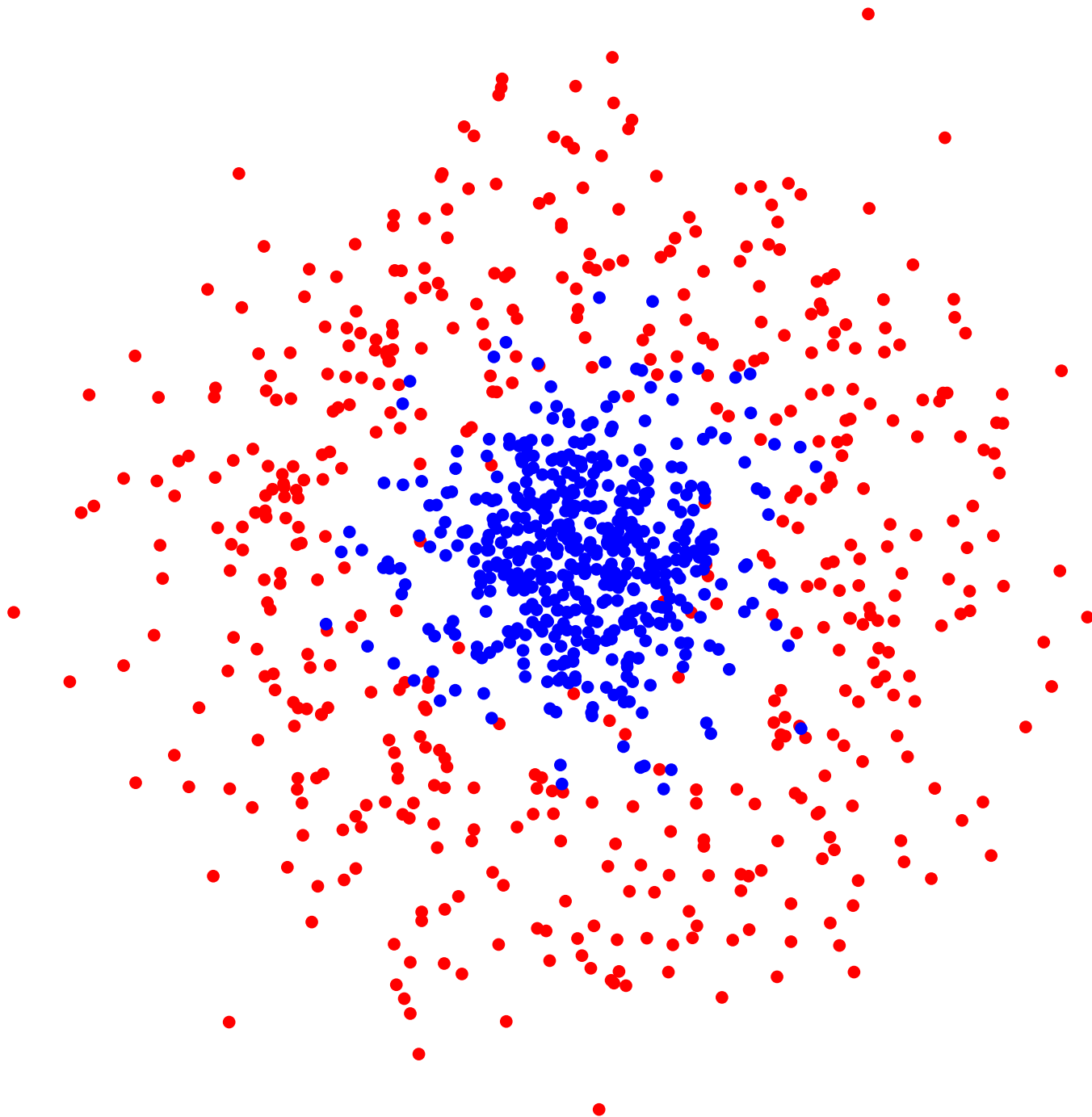


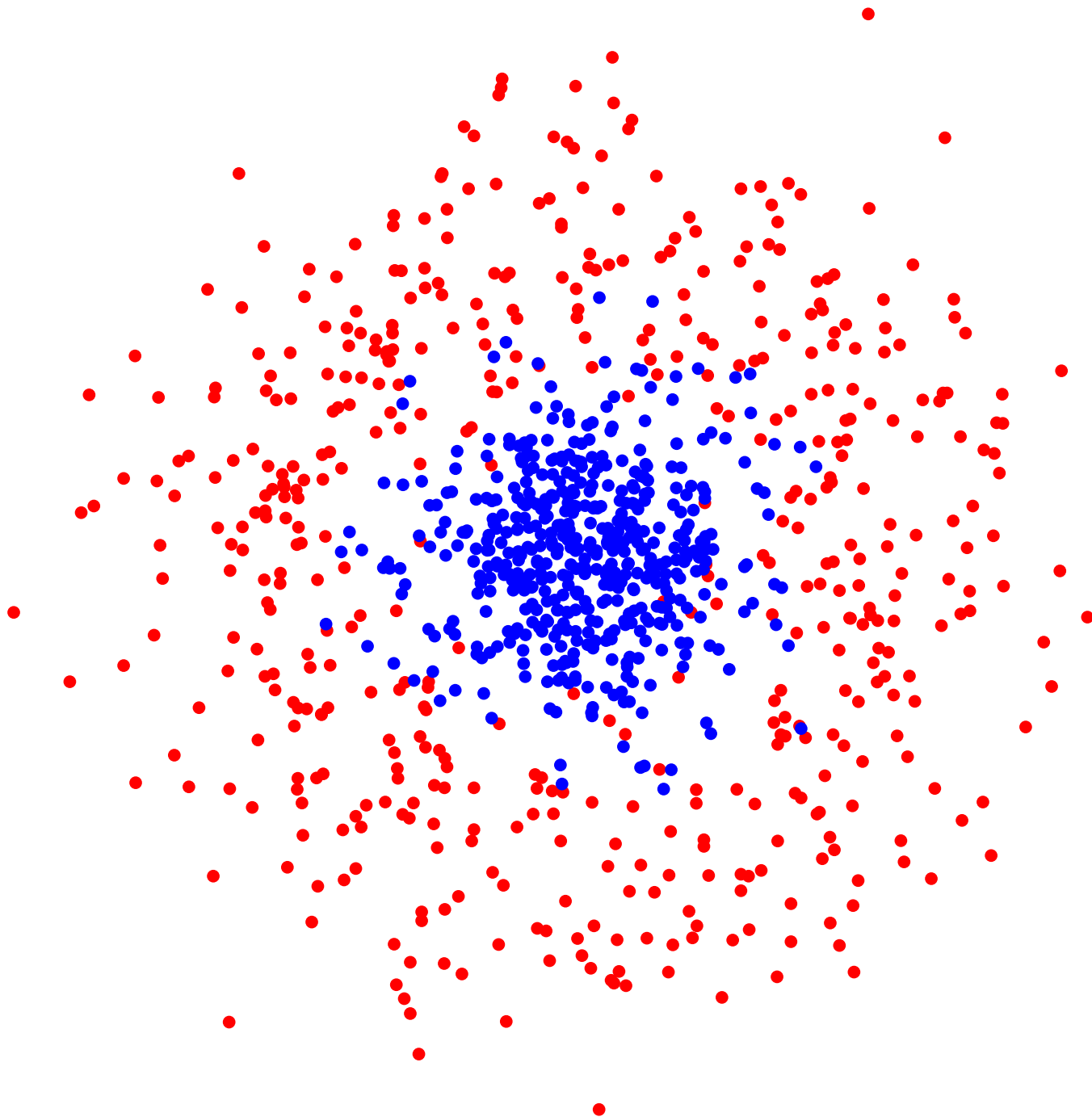
..... -1

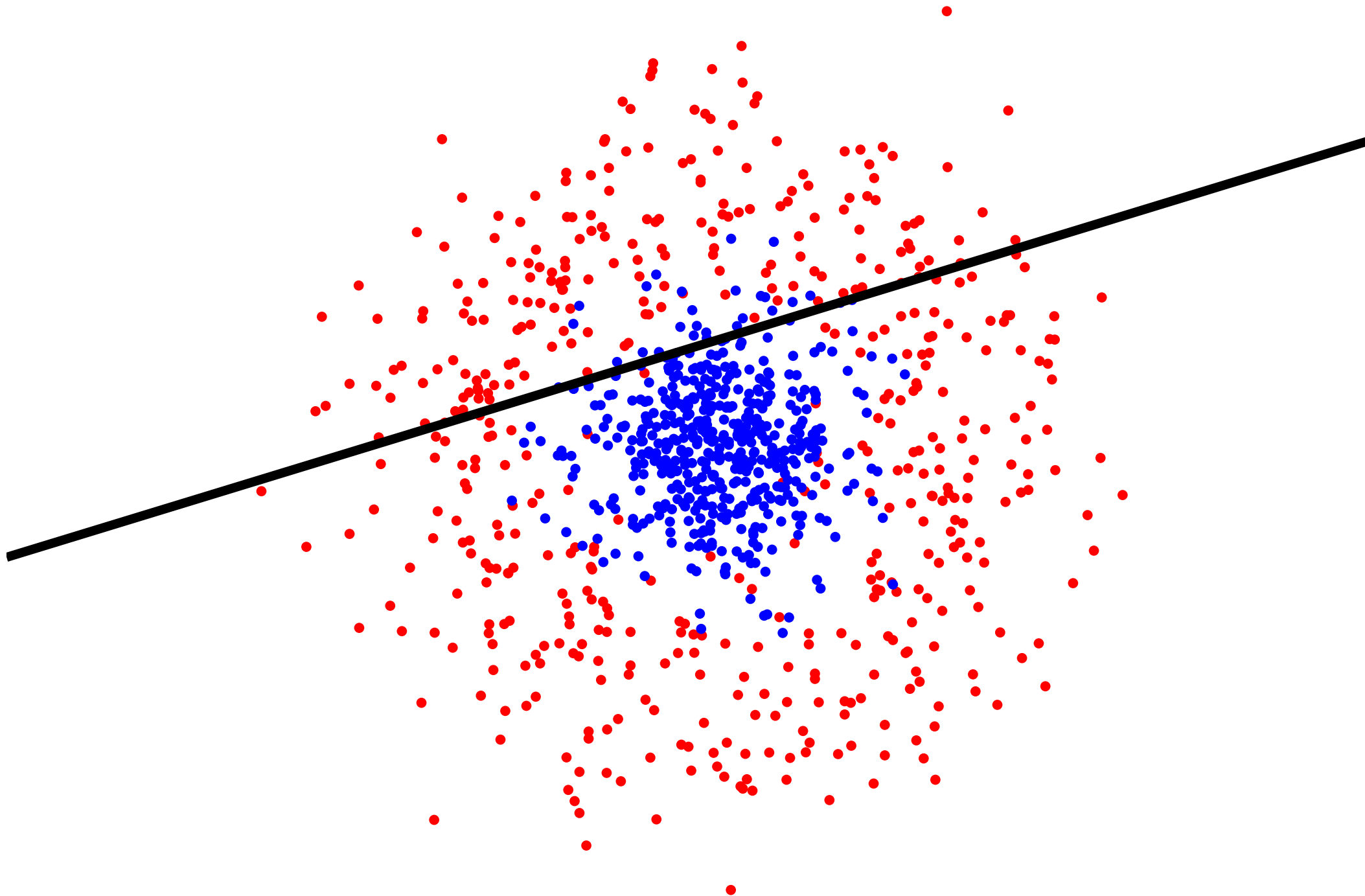


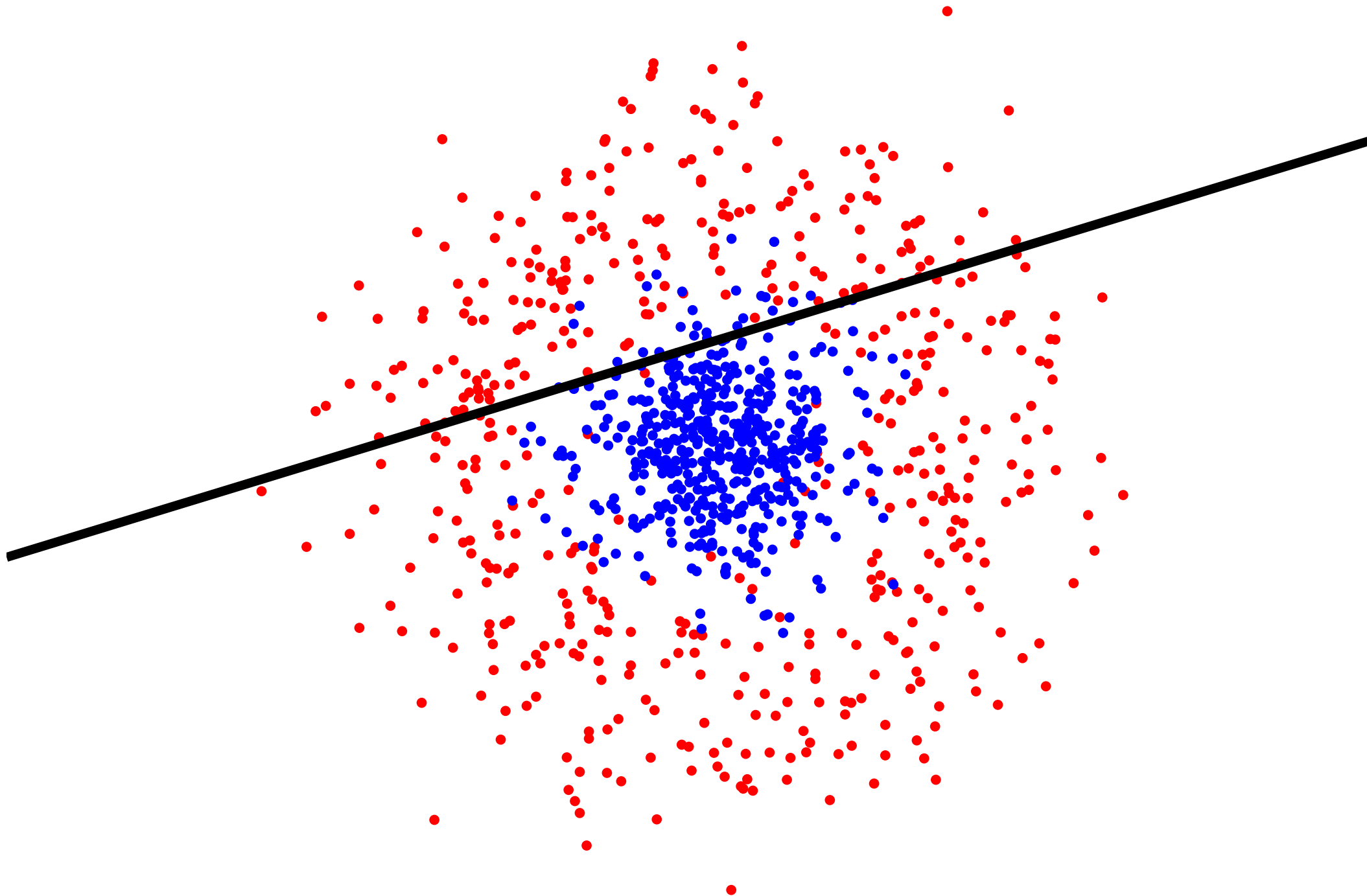
..... +1

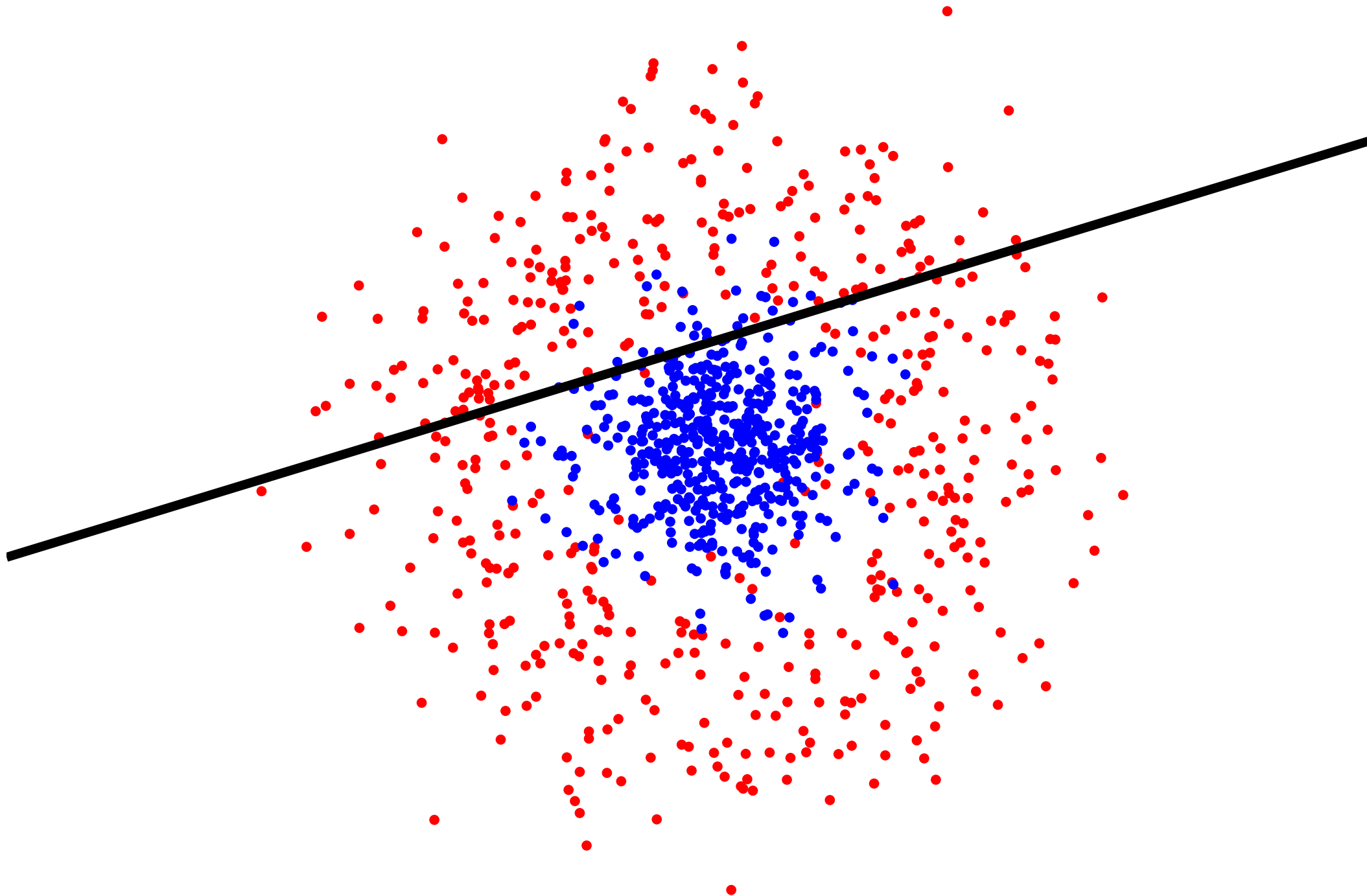


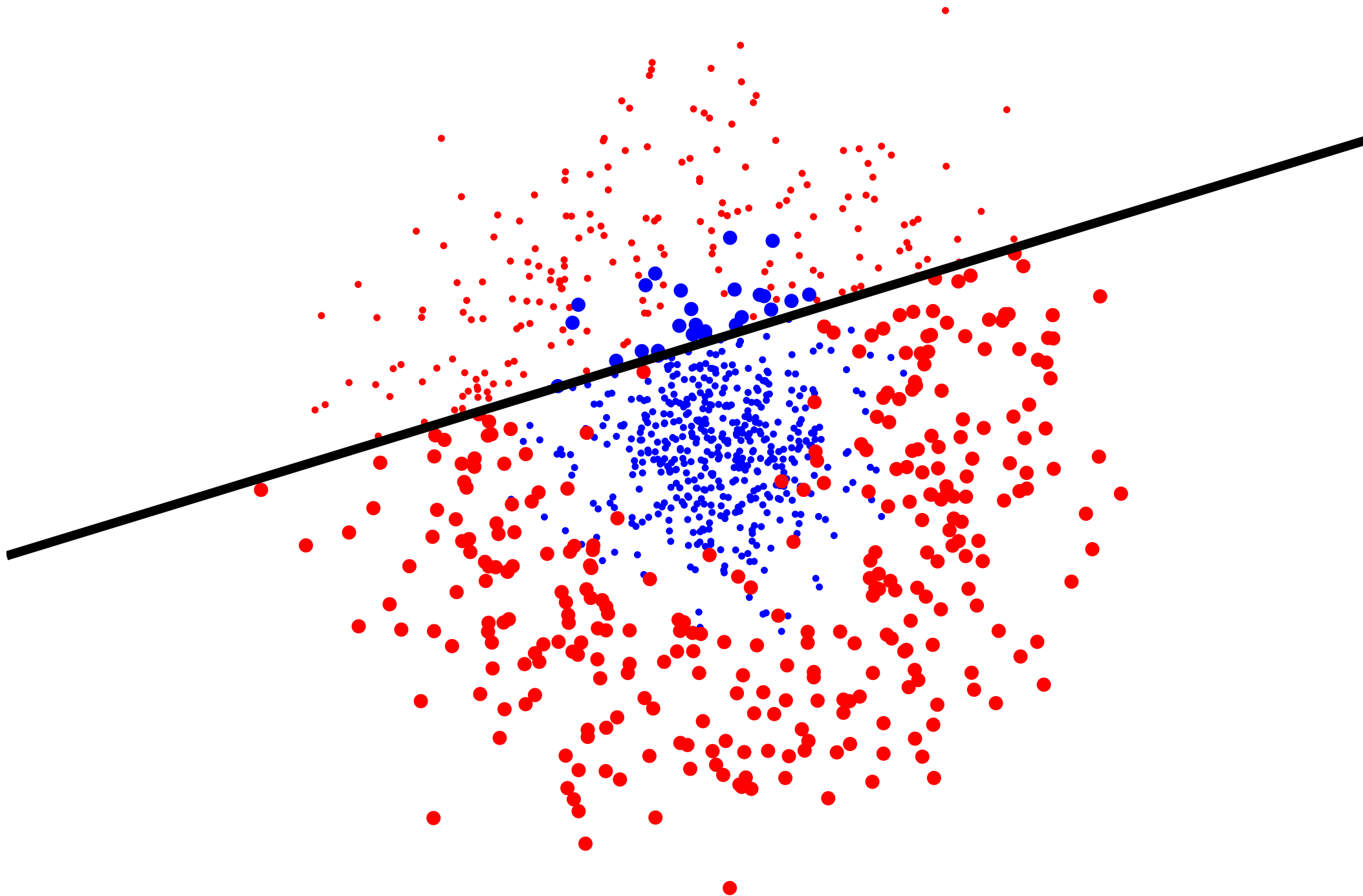


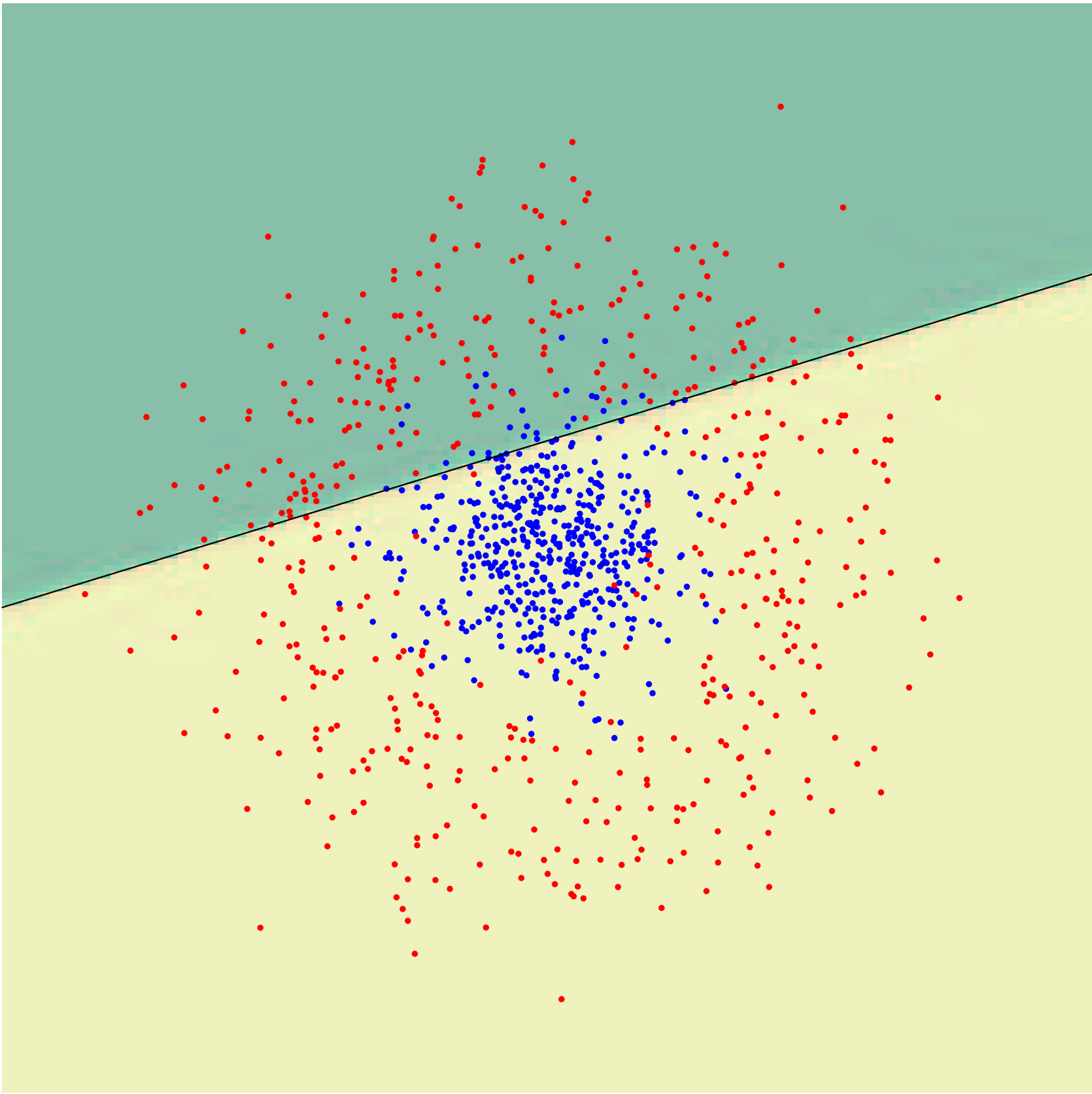


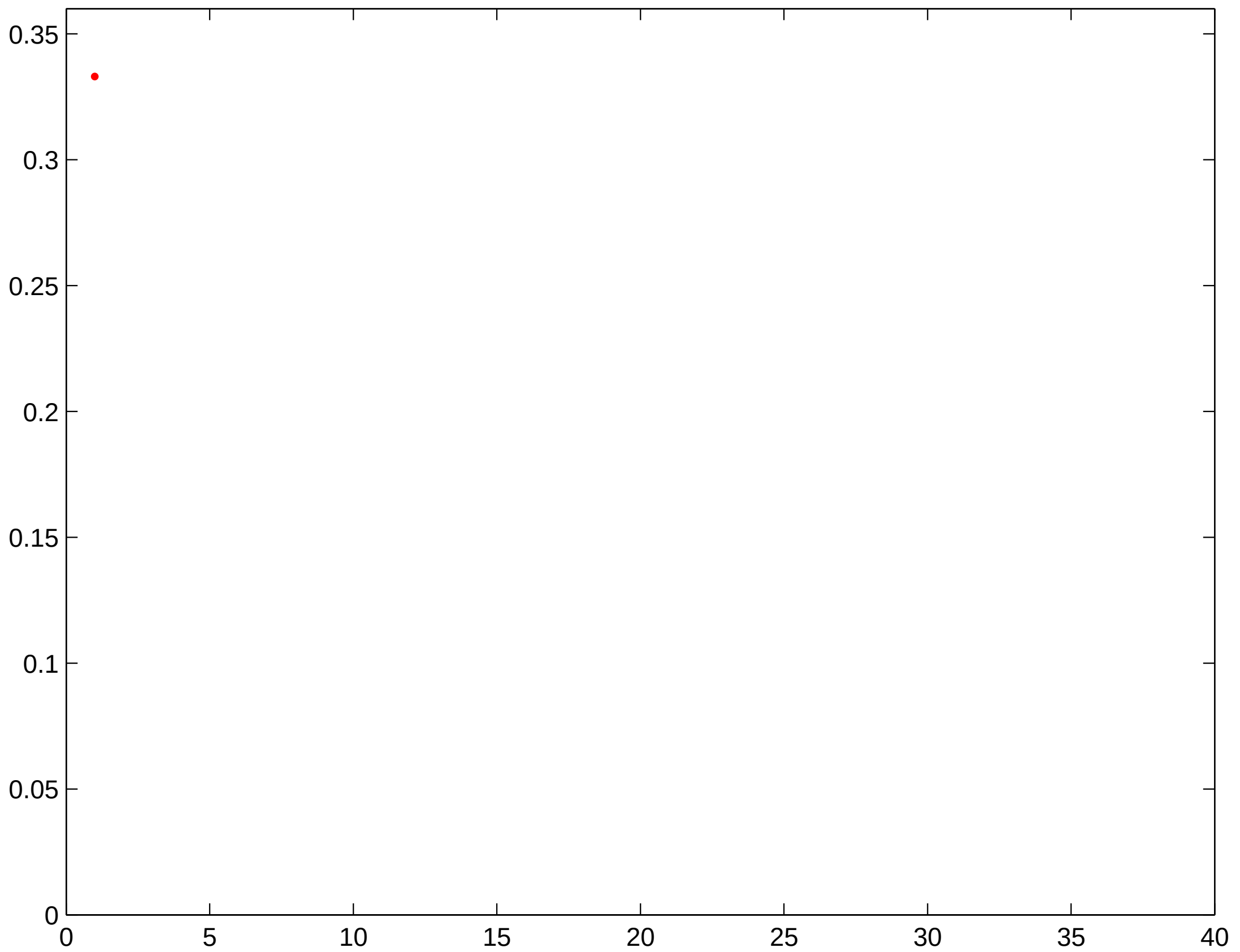


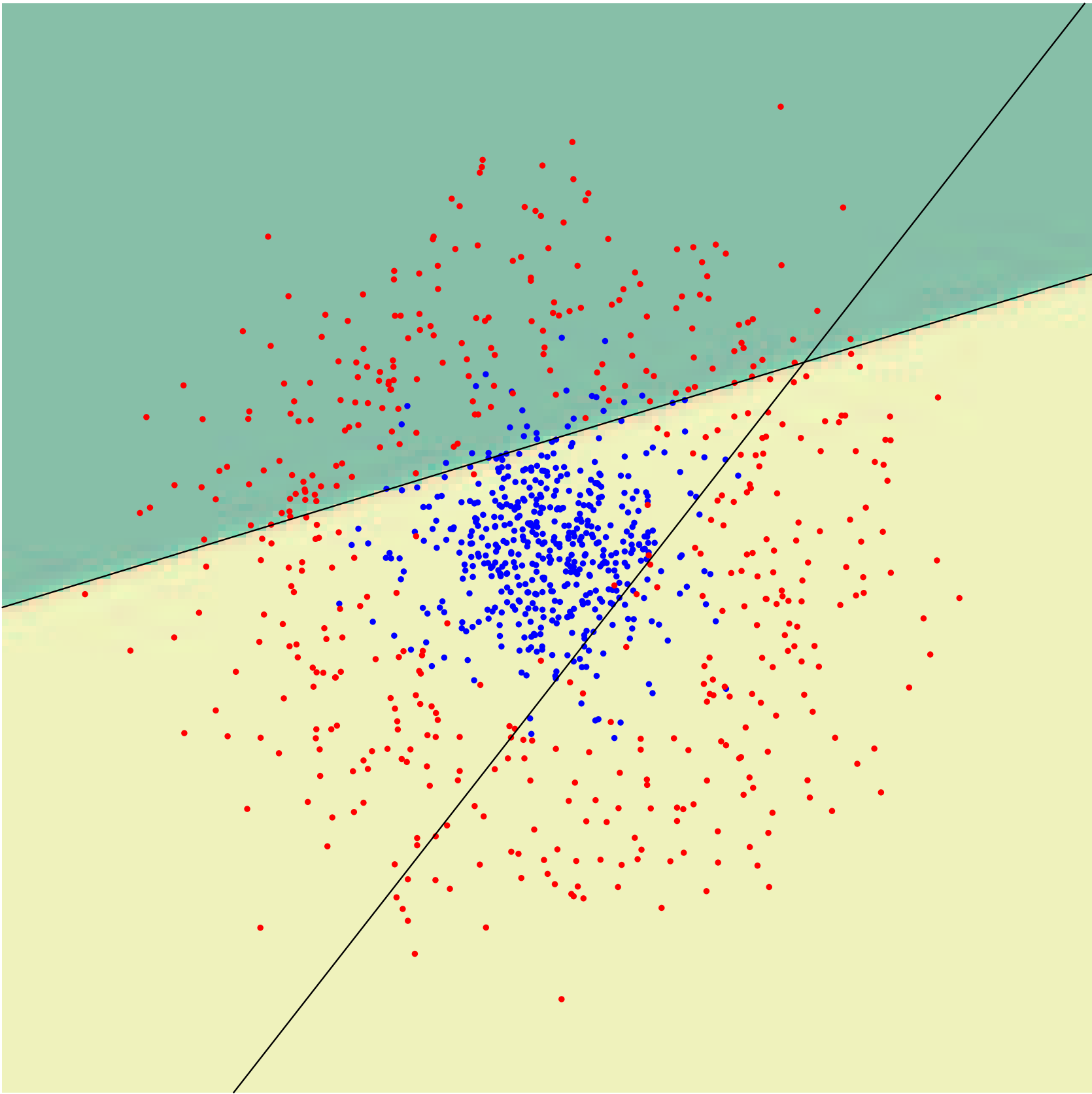


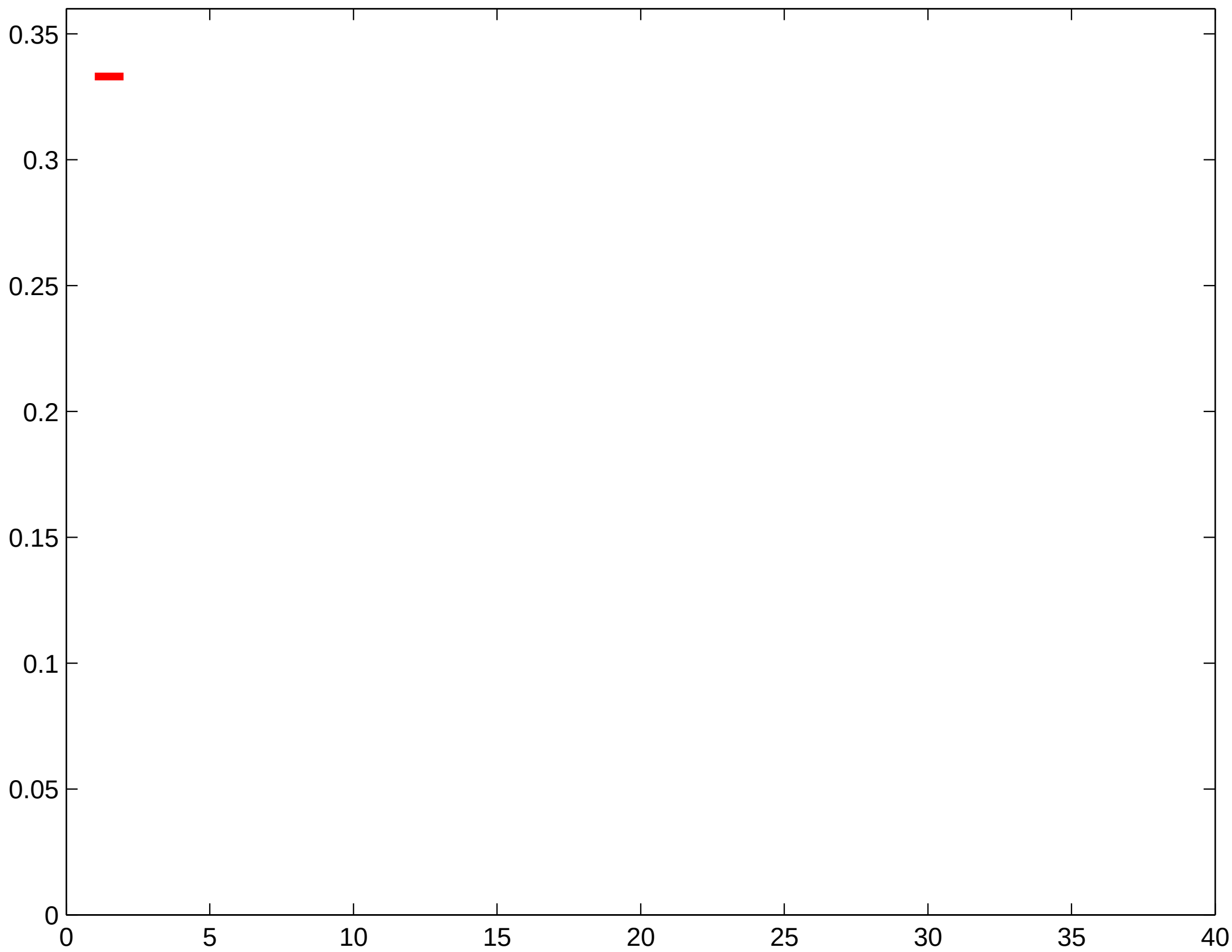


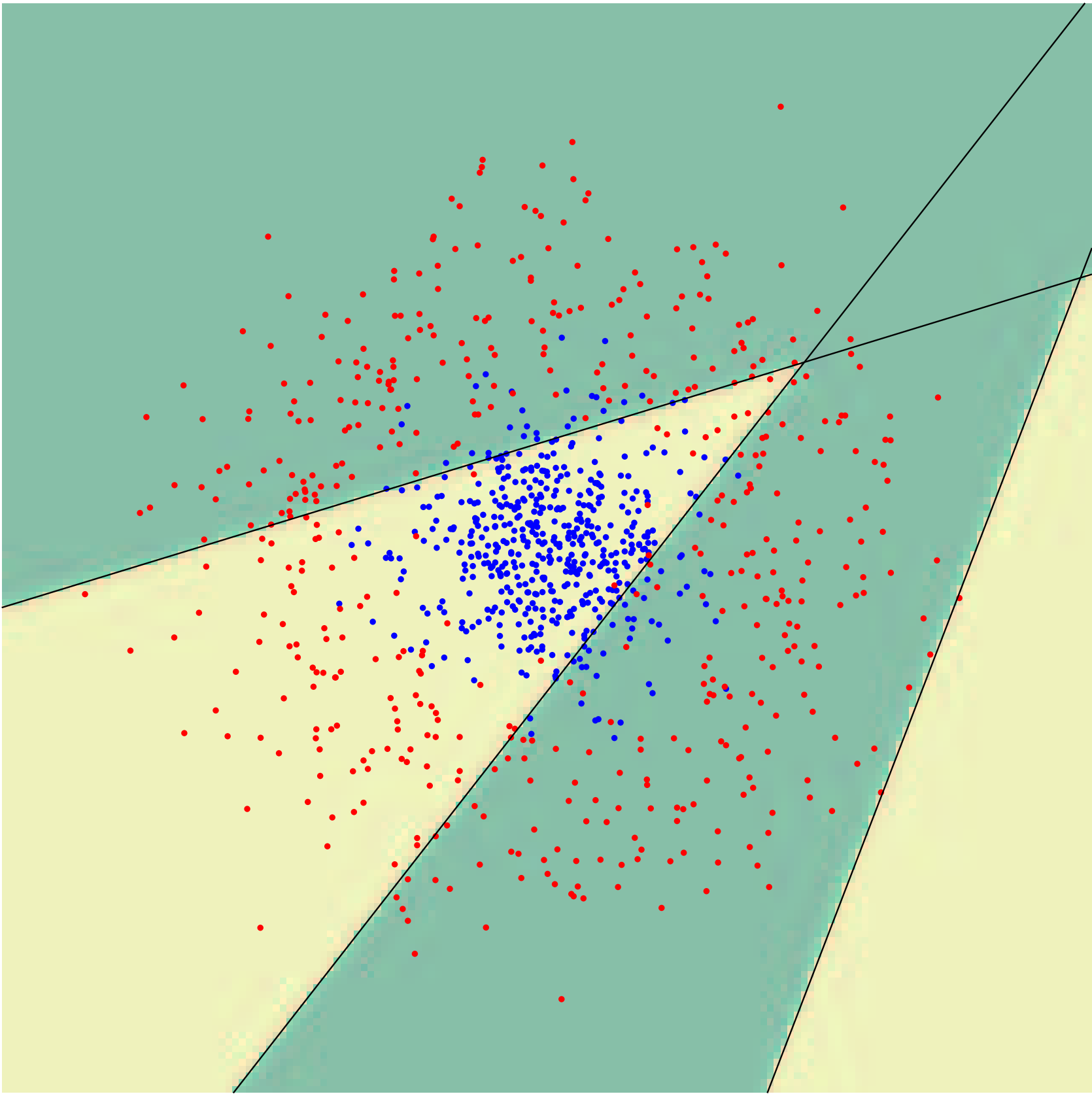


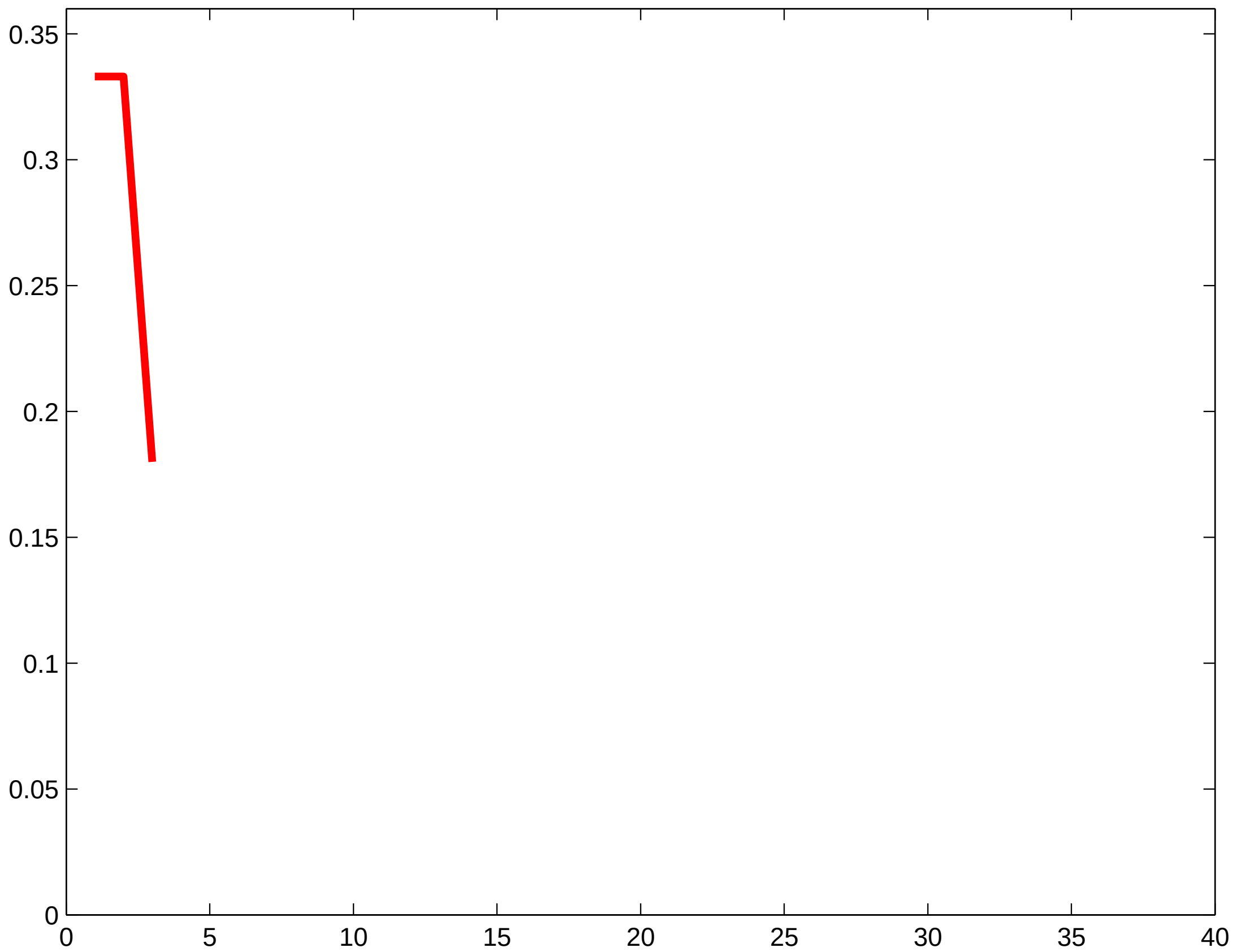


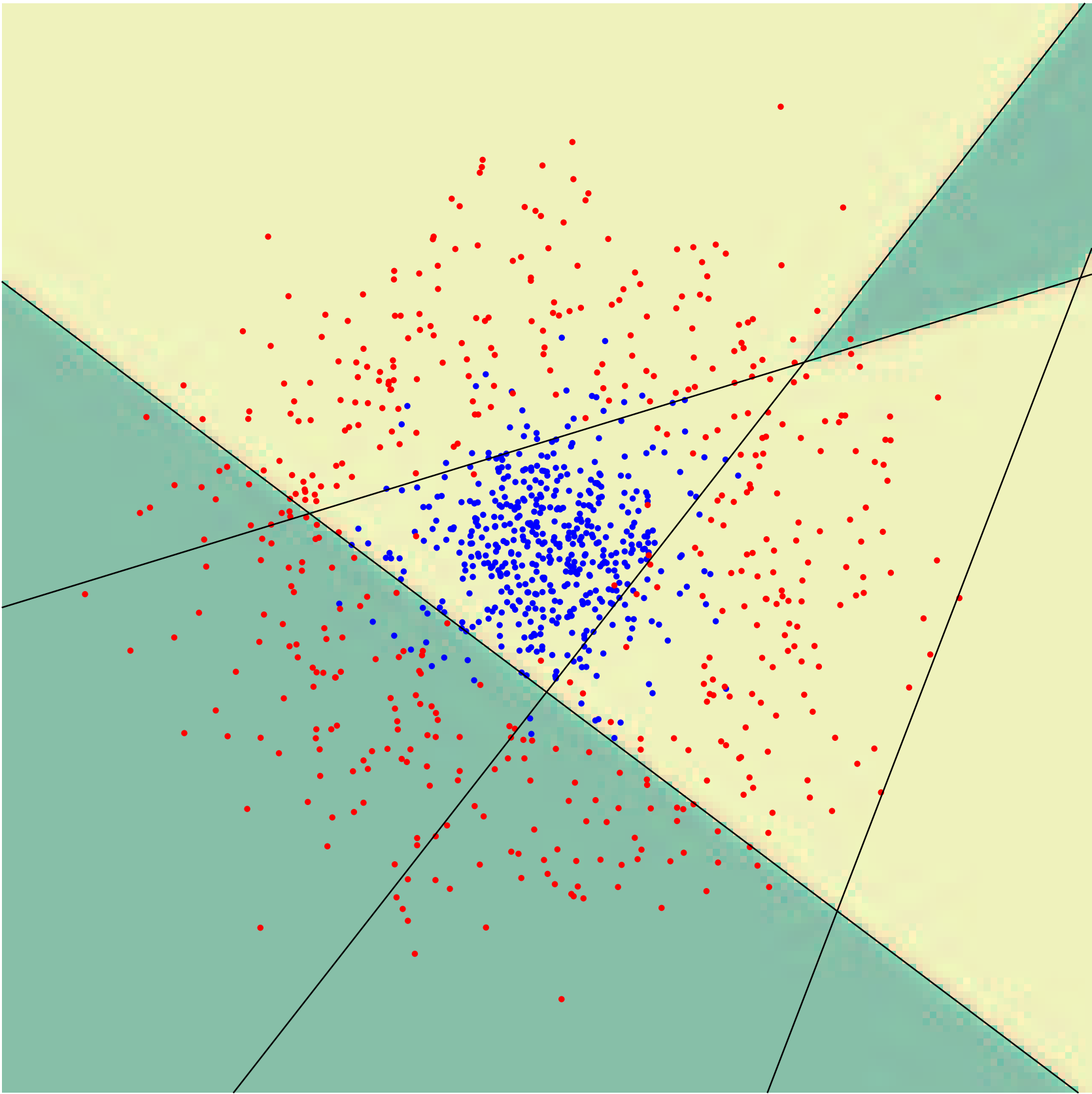


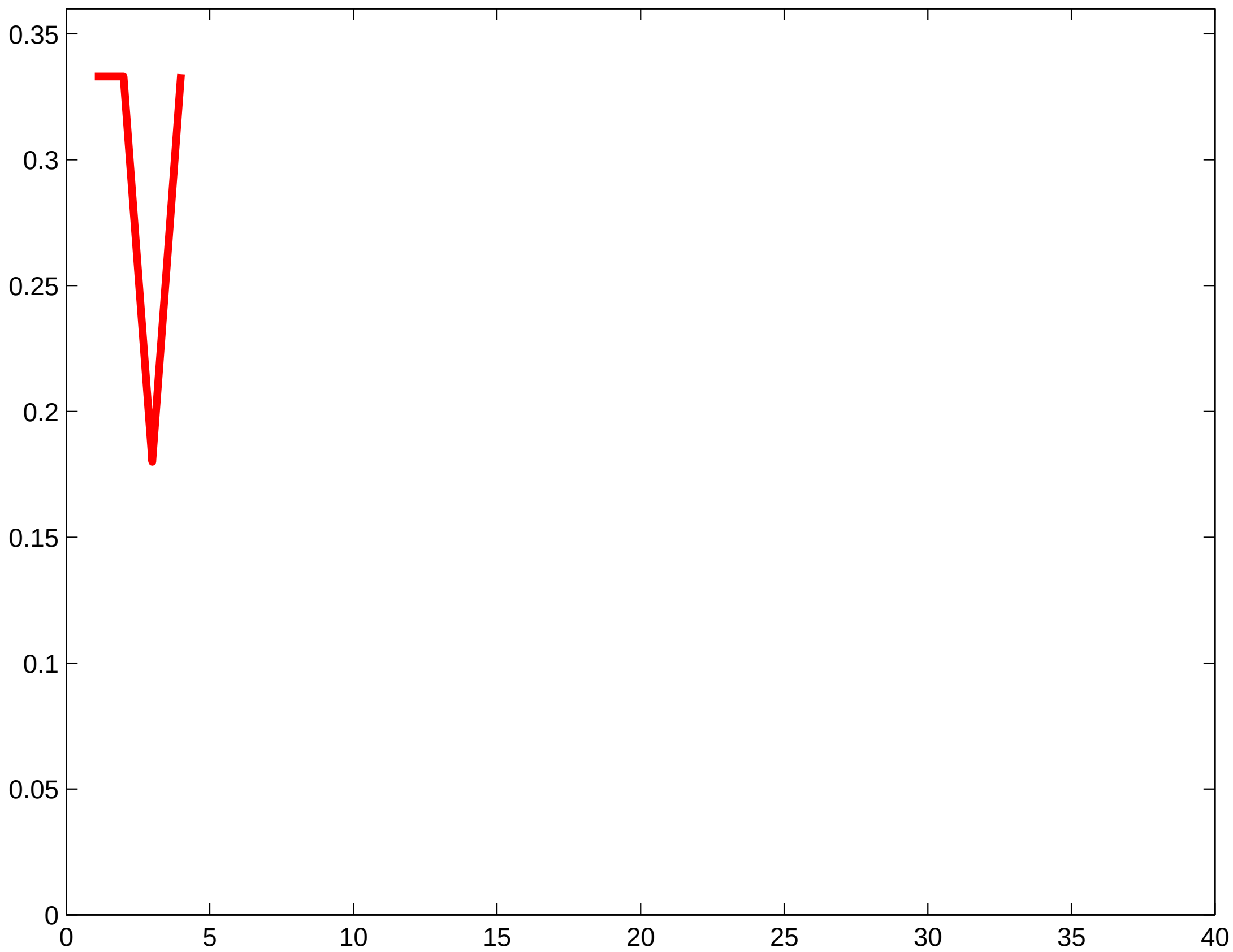


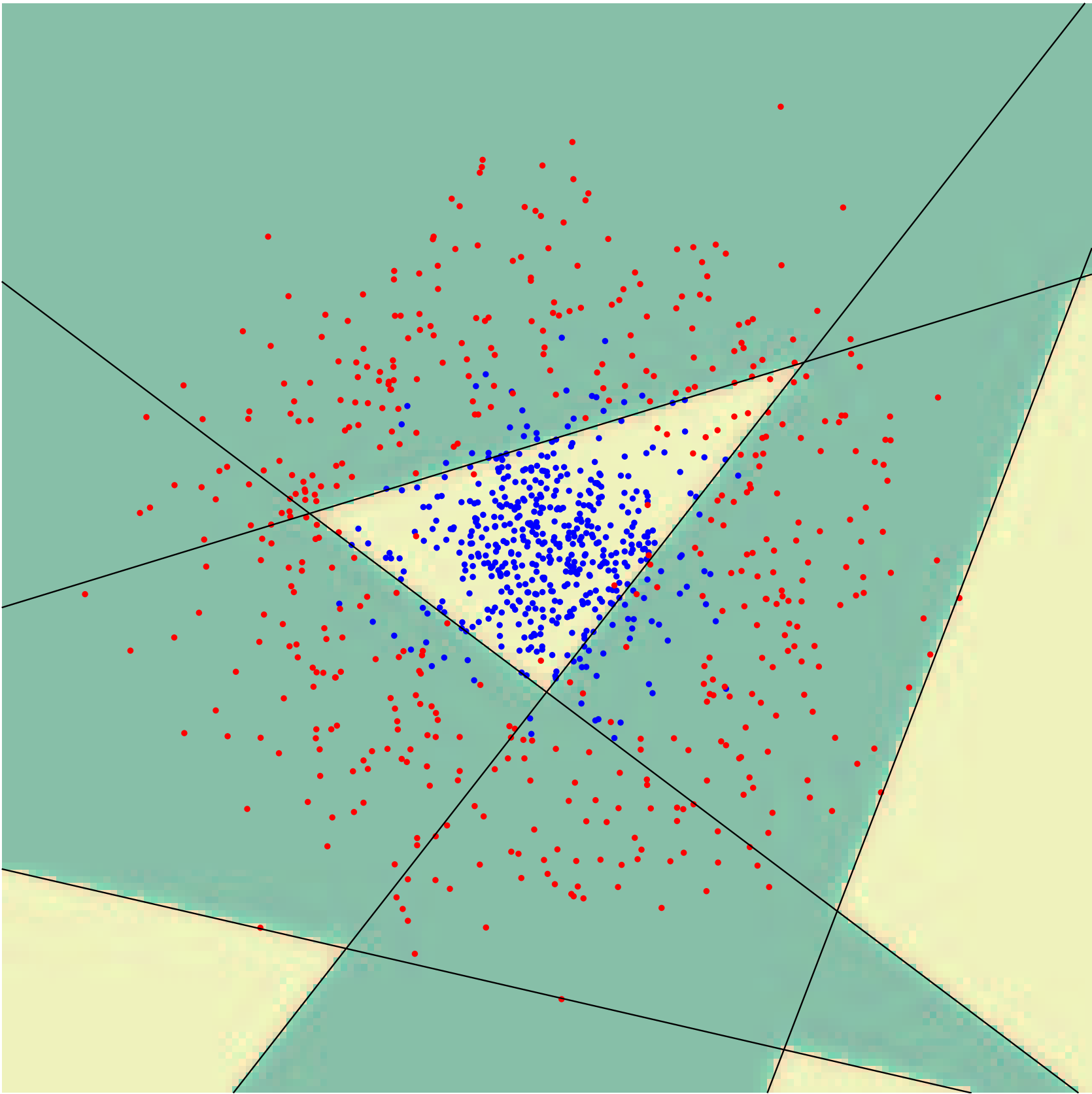


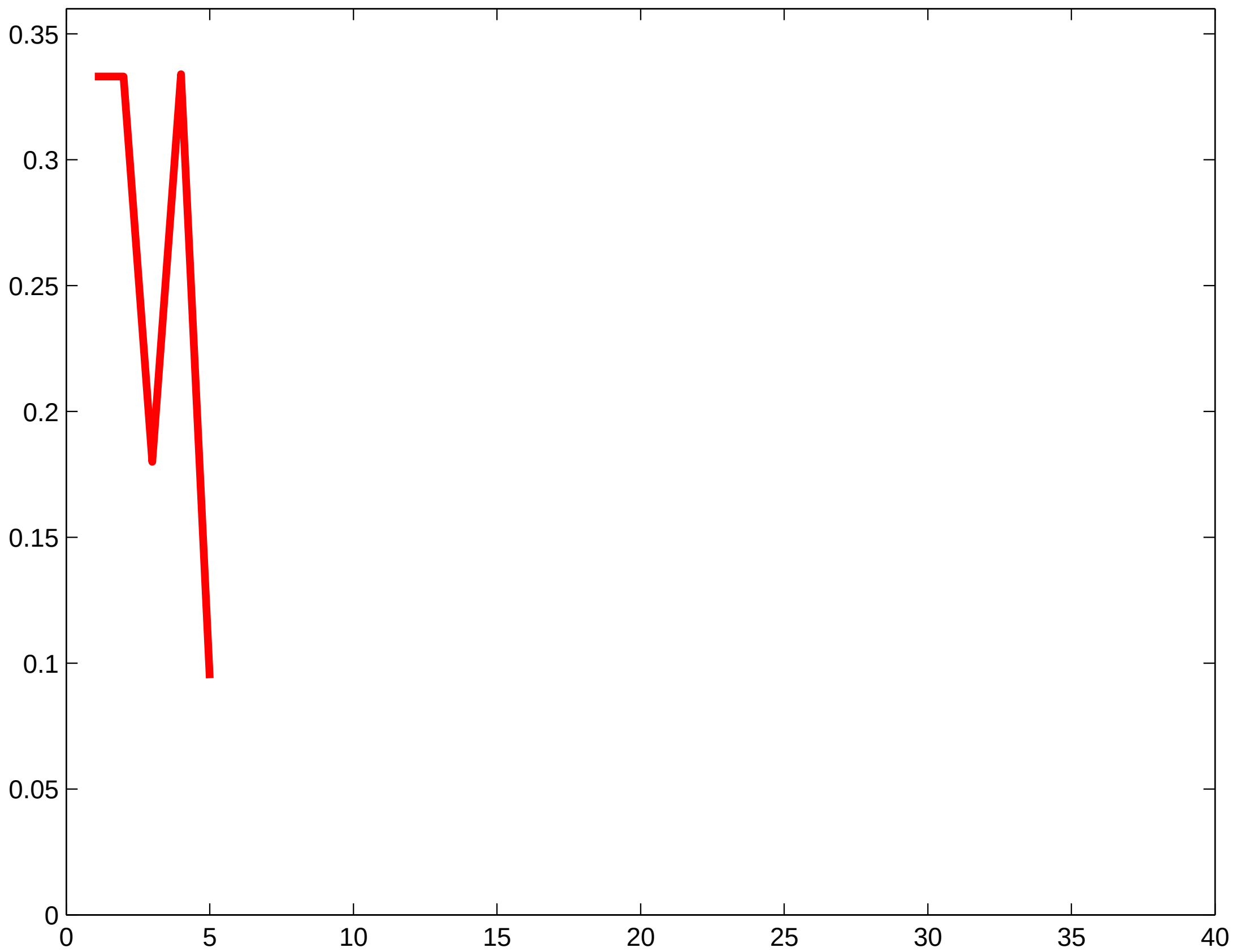


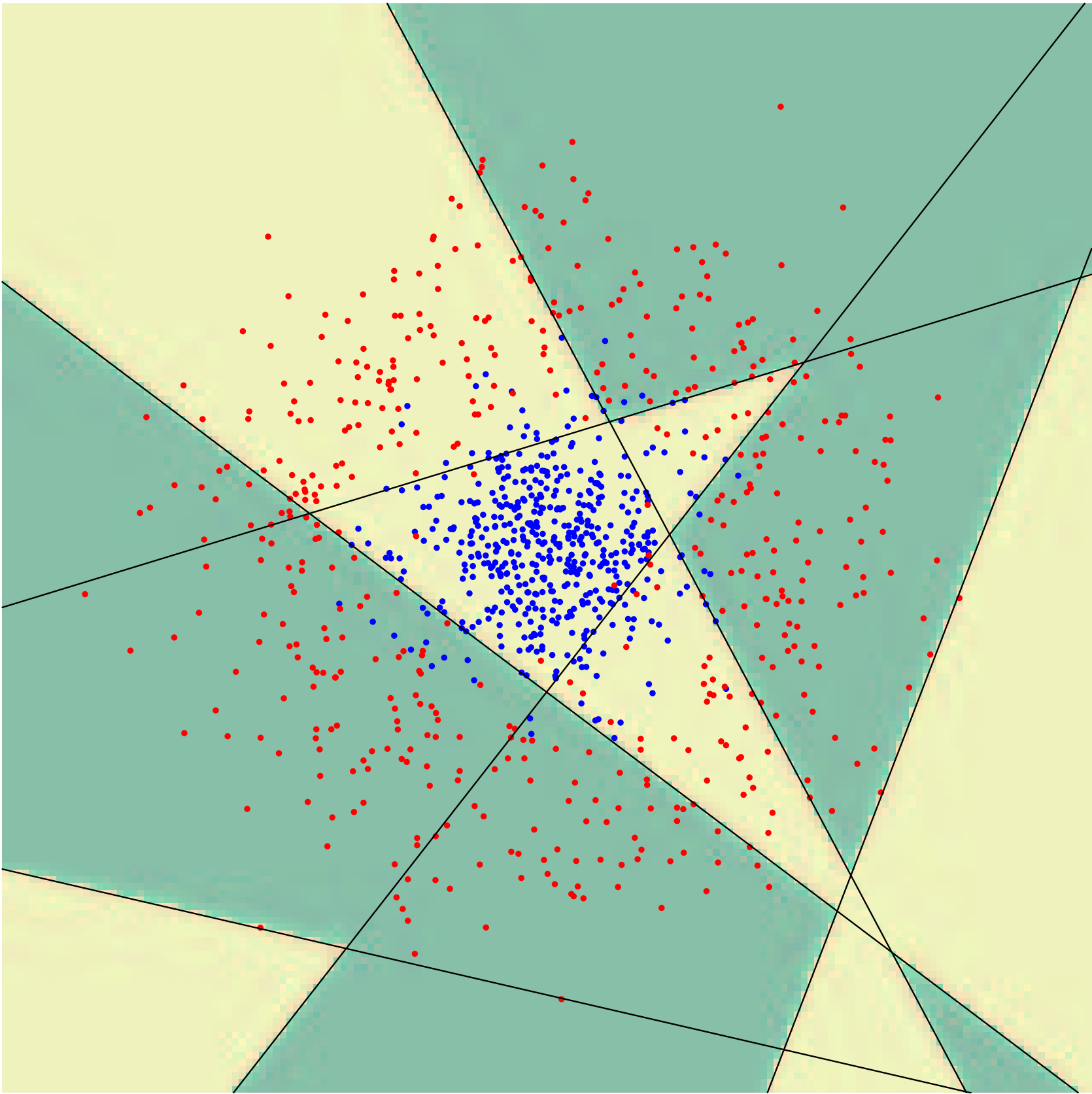


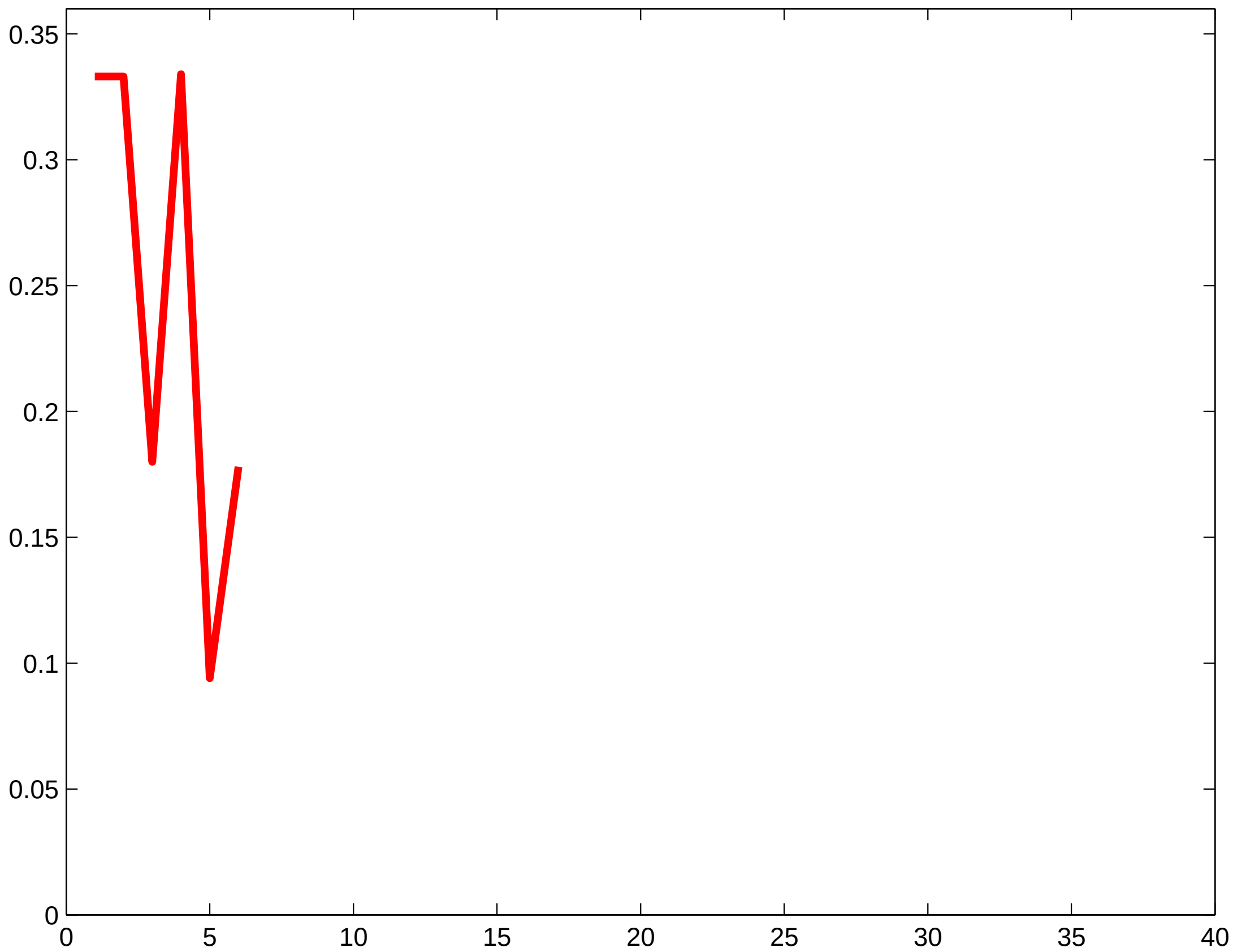


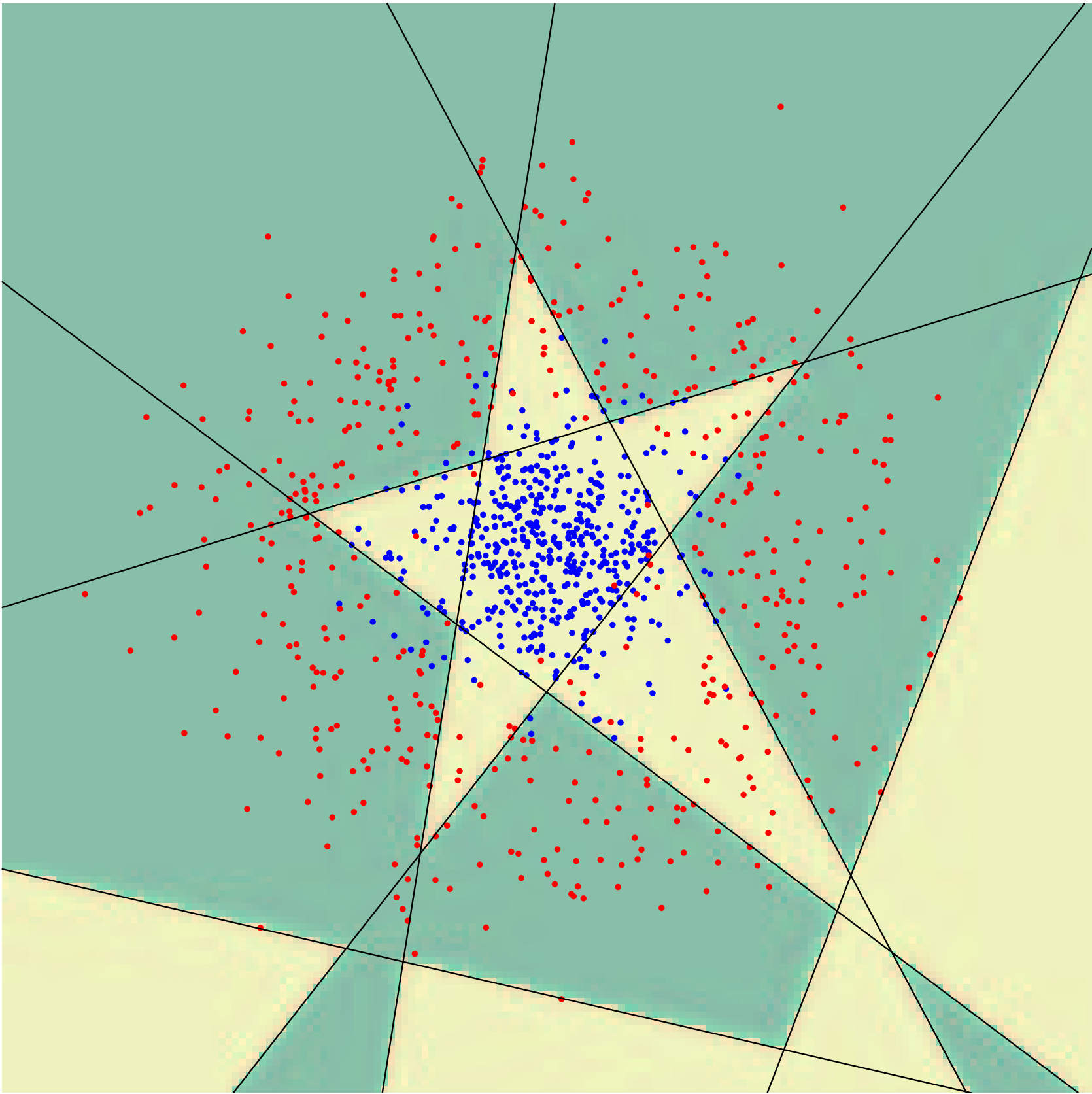


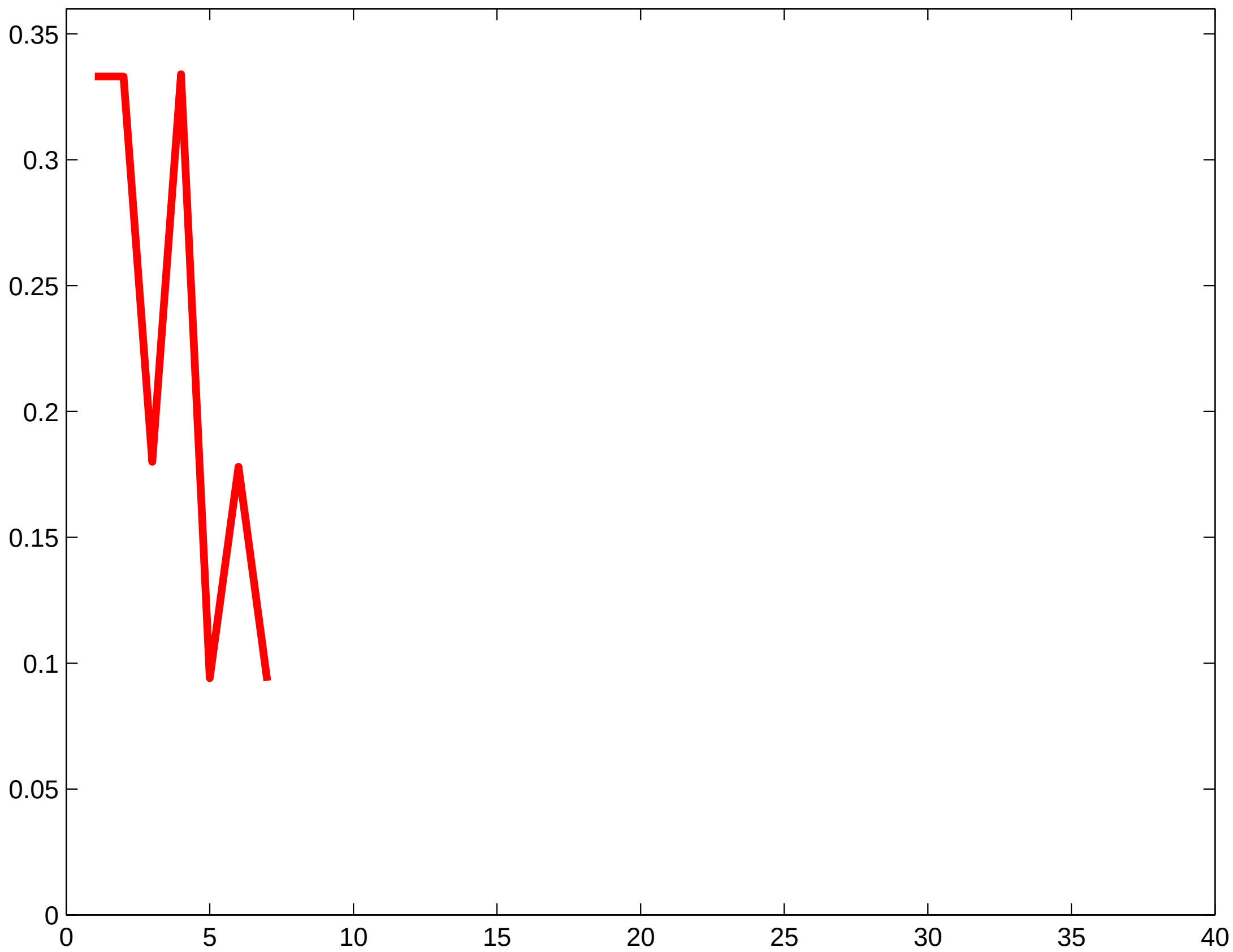


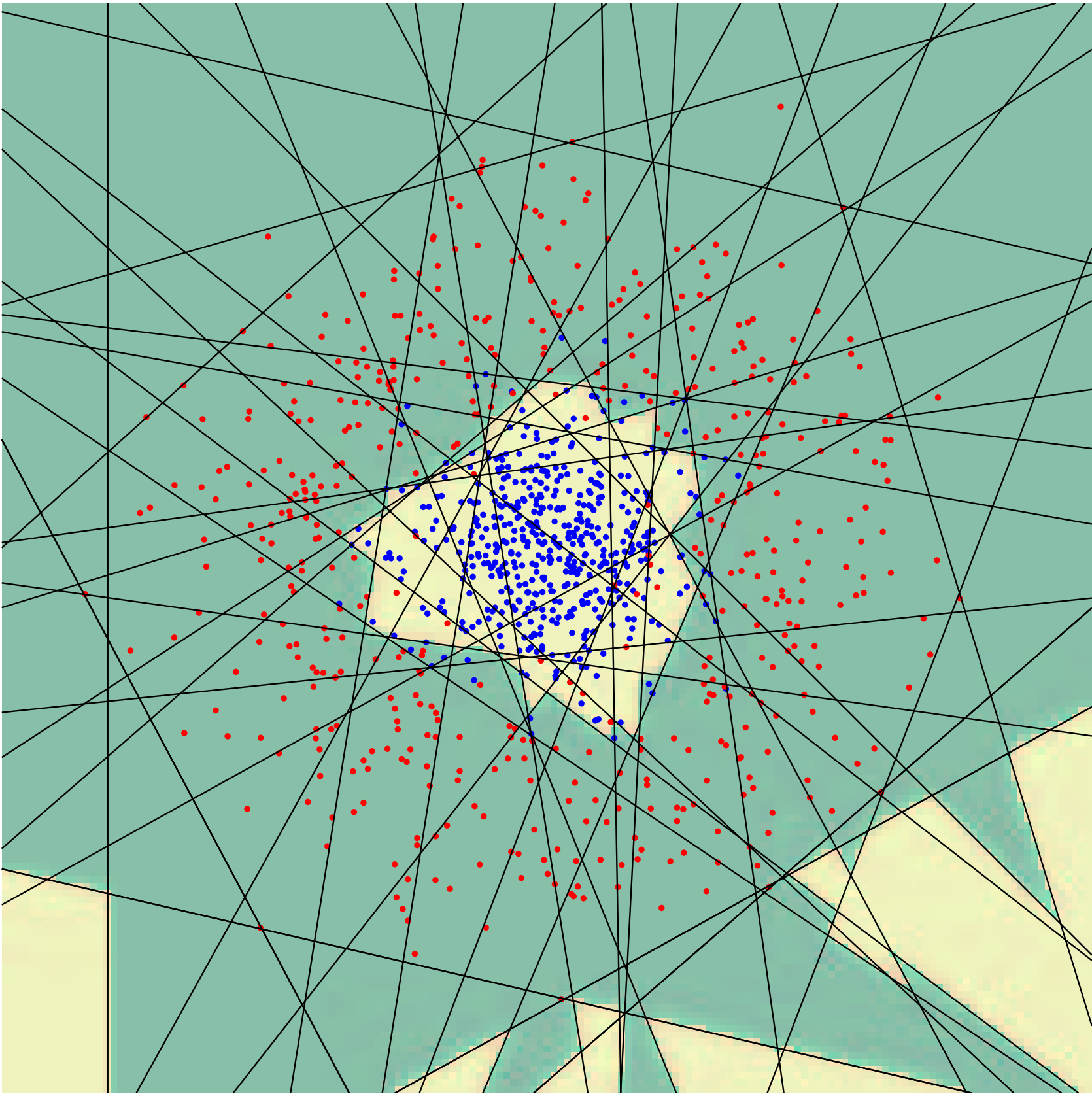


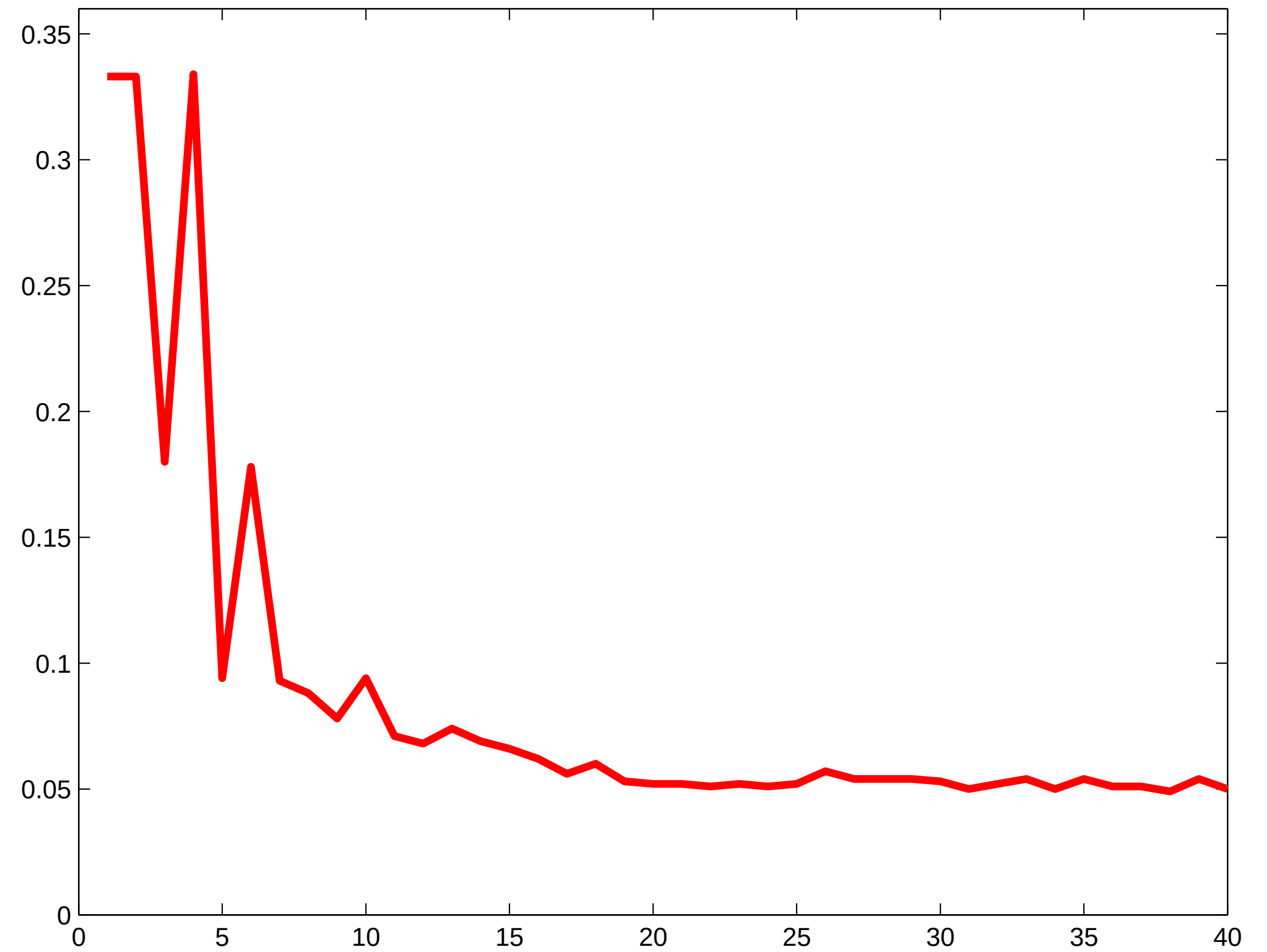


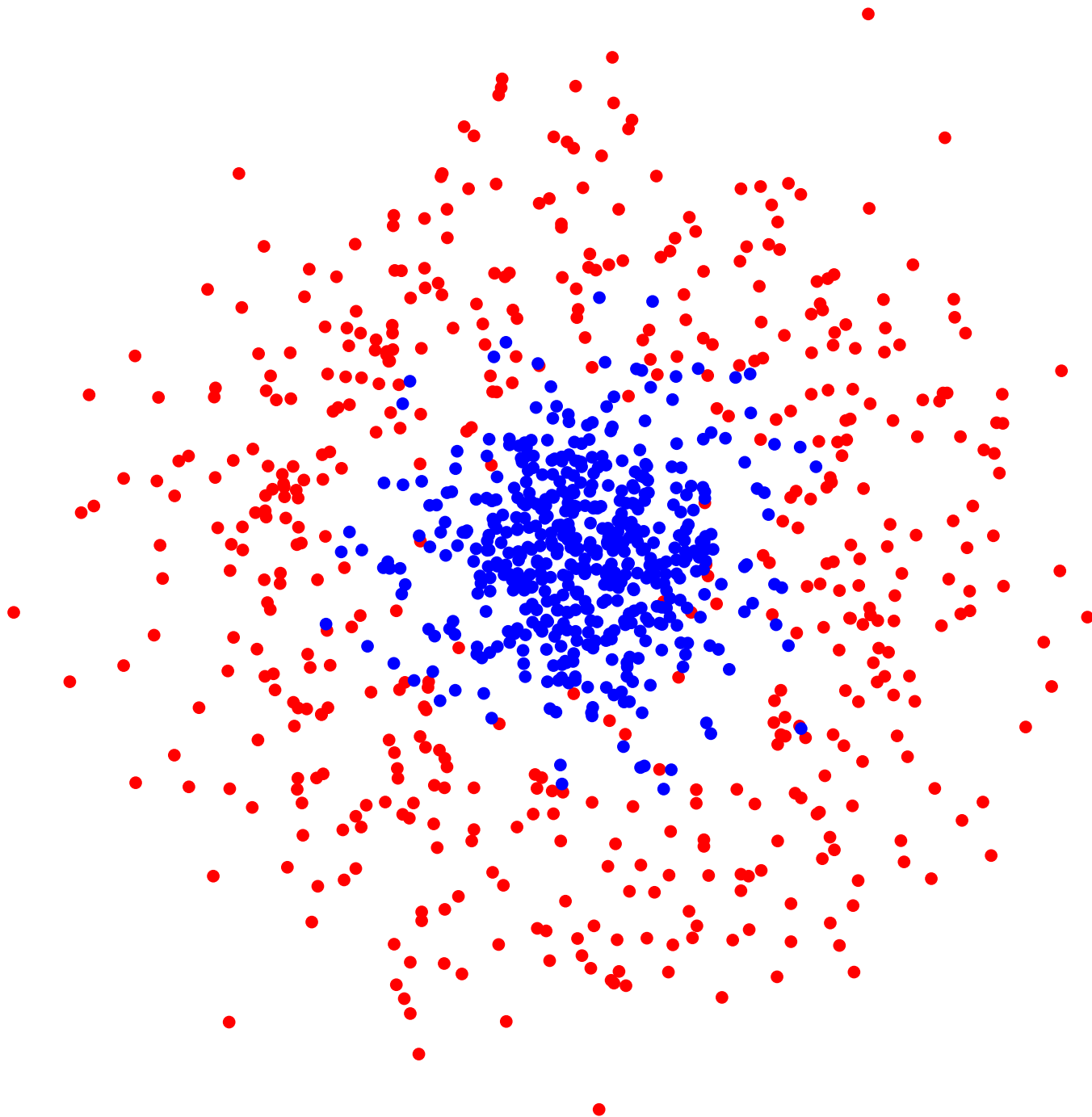


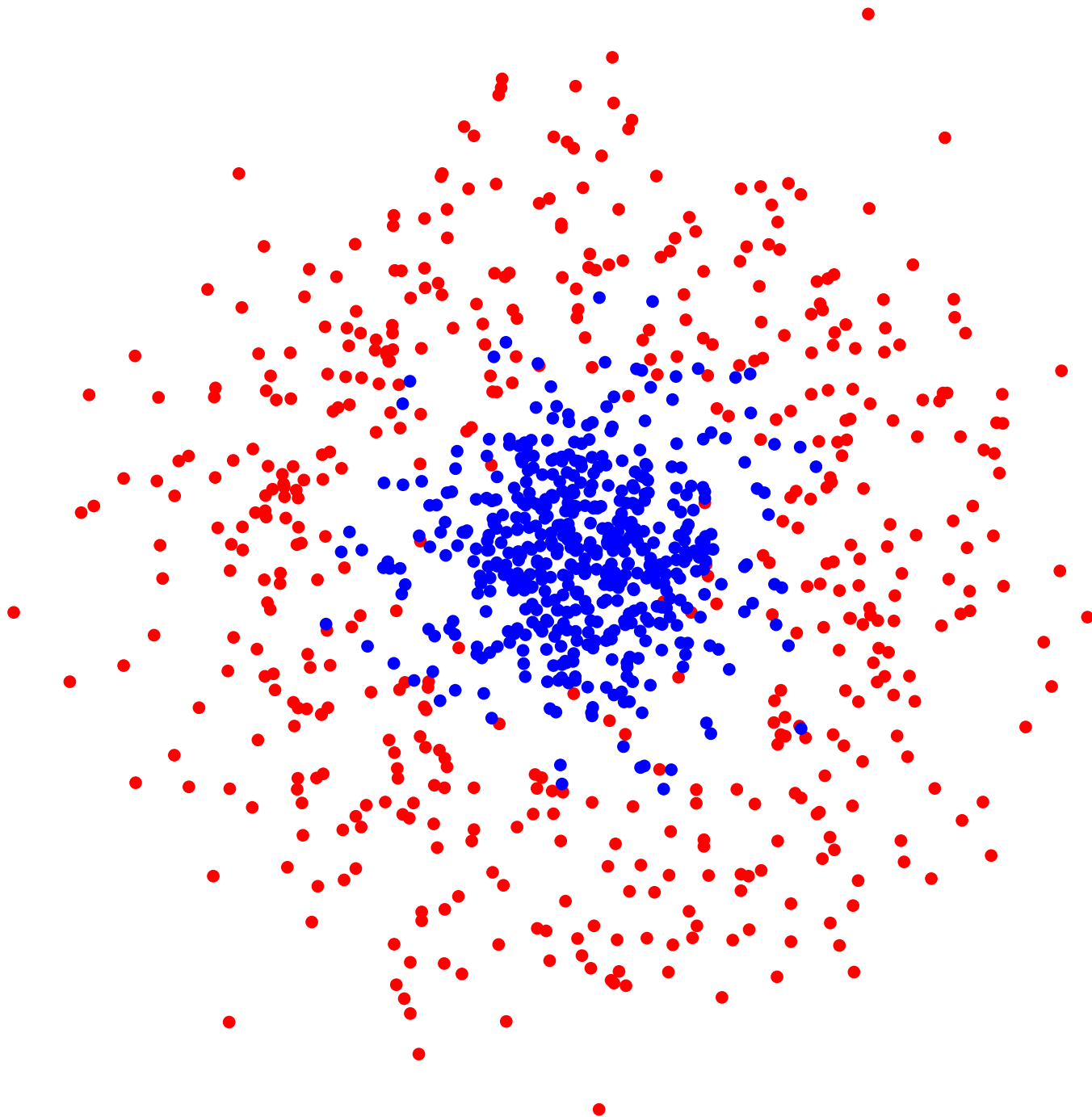


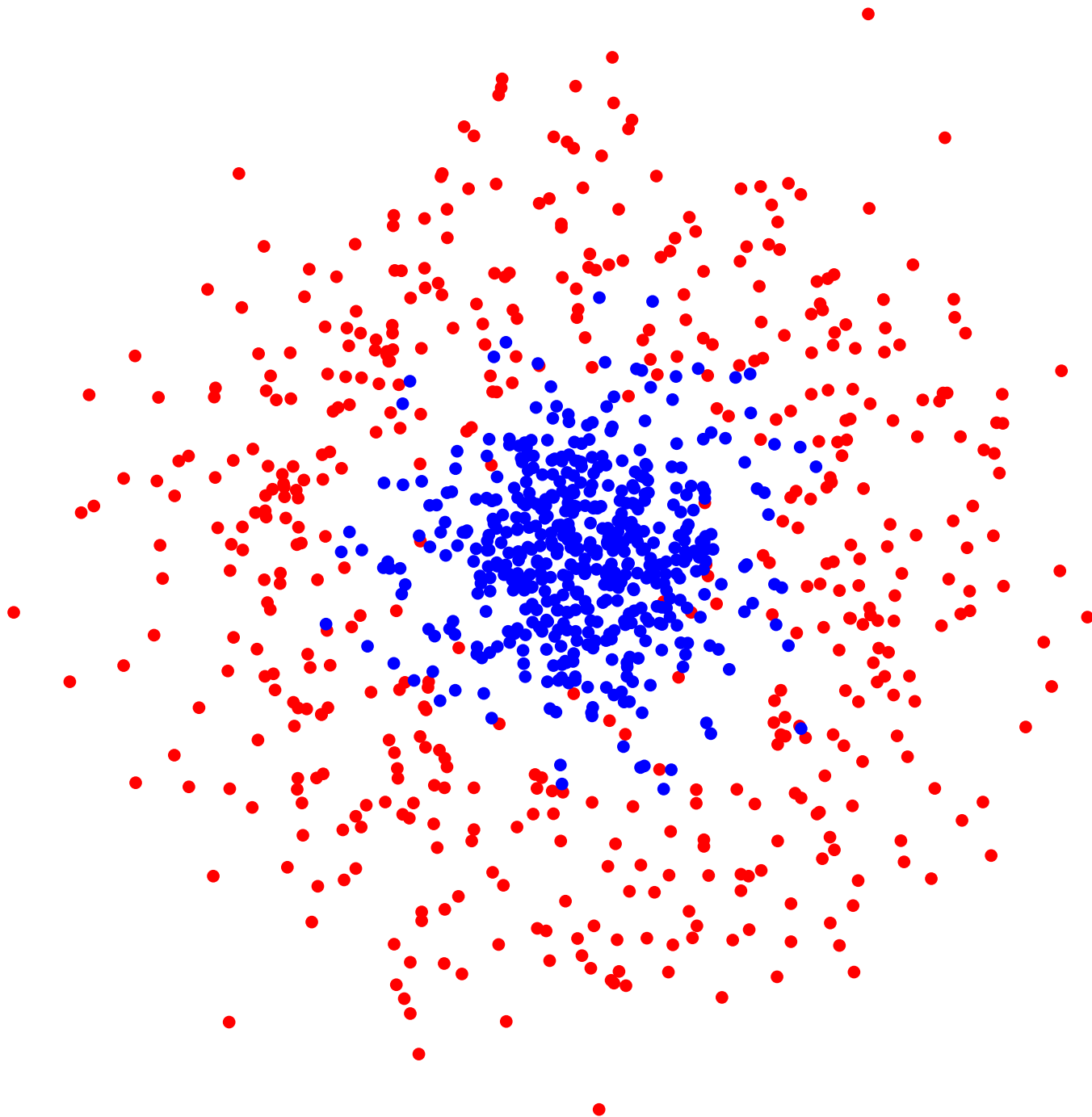


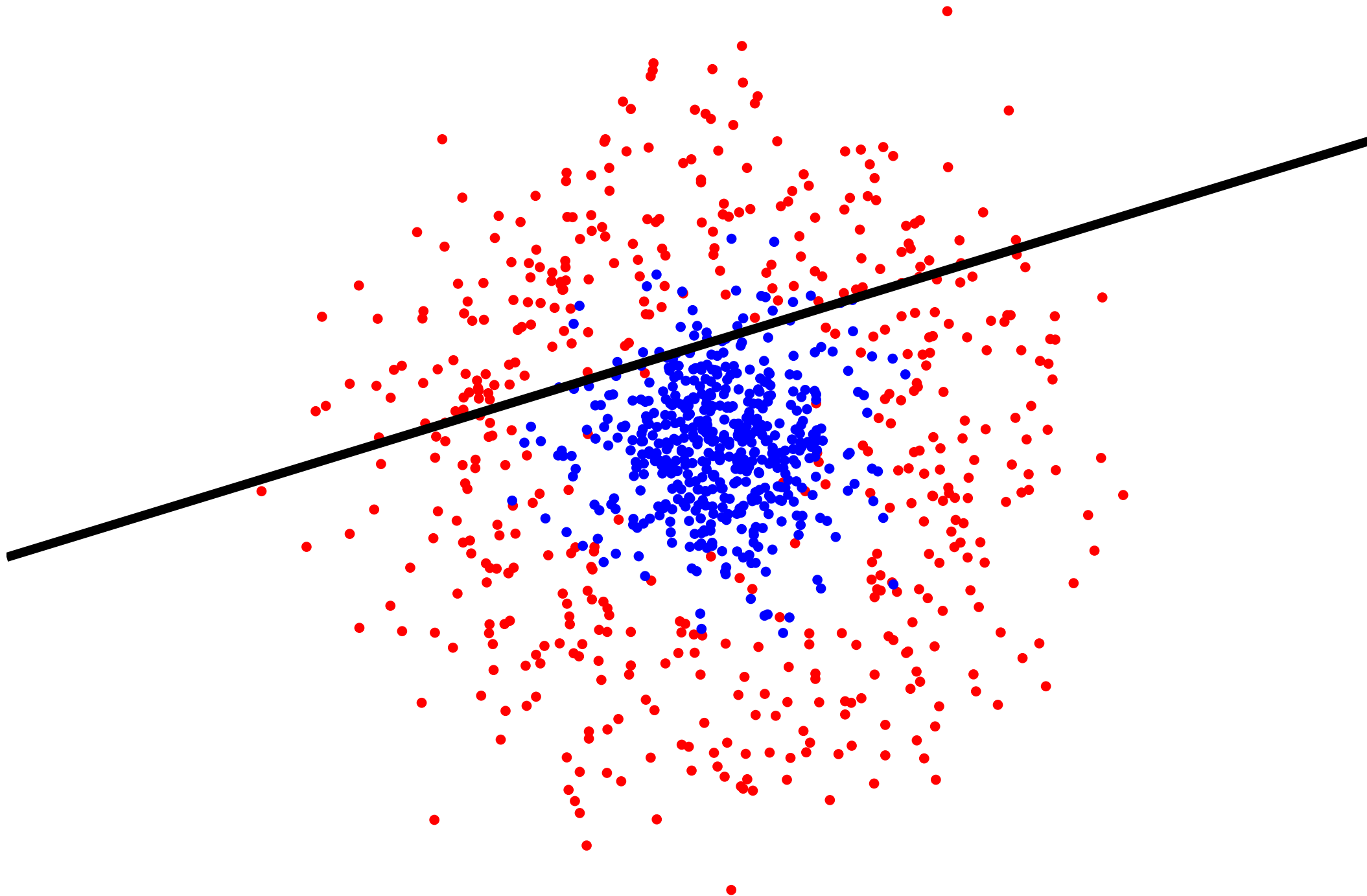


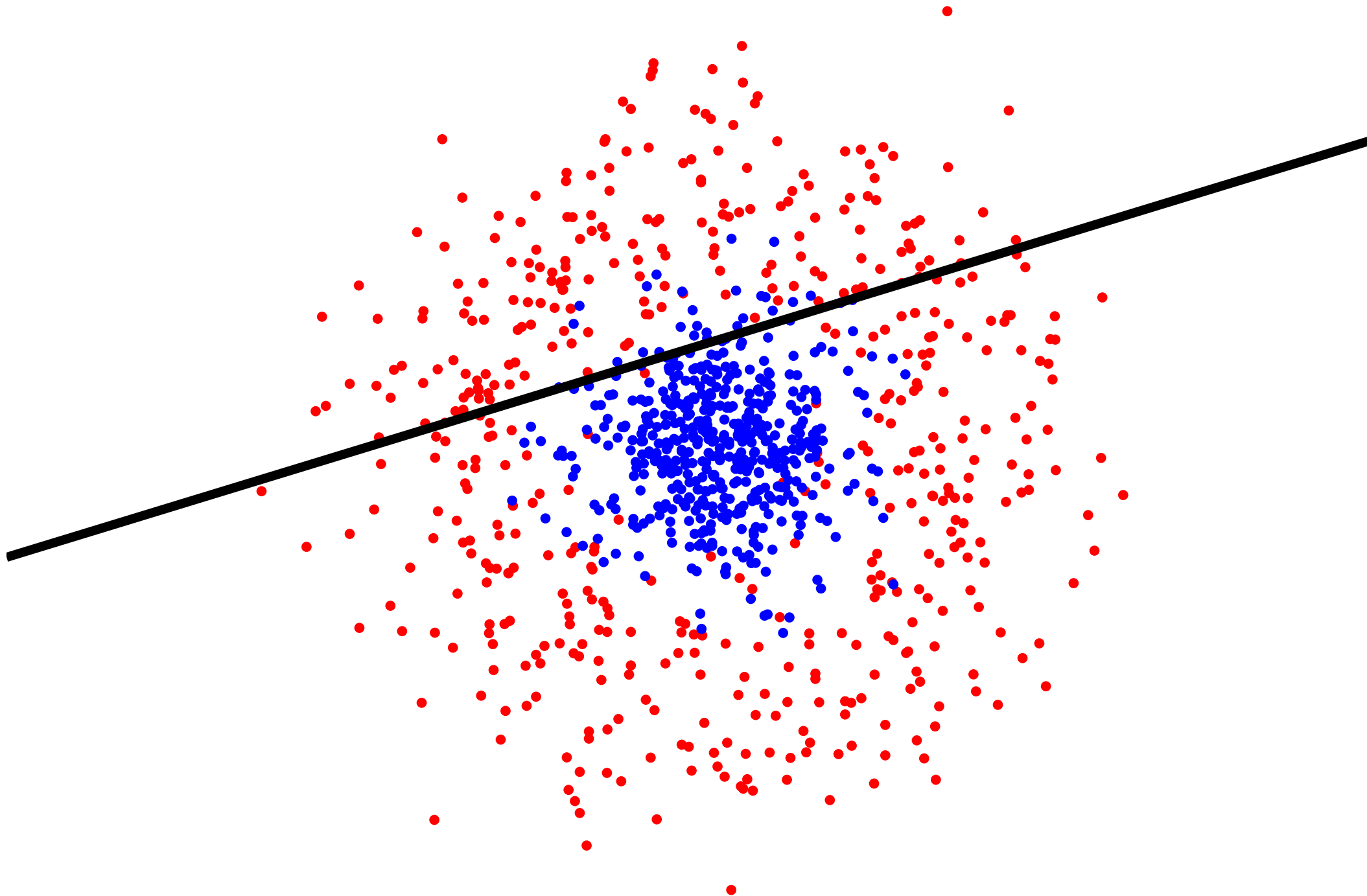


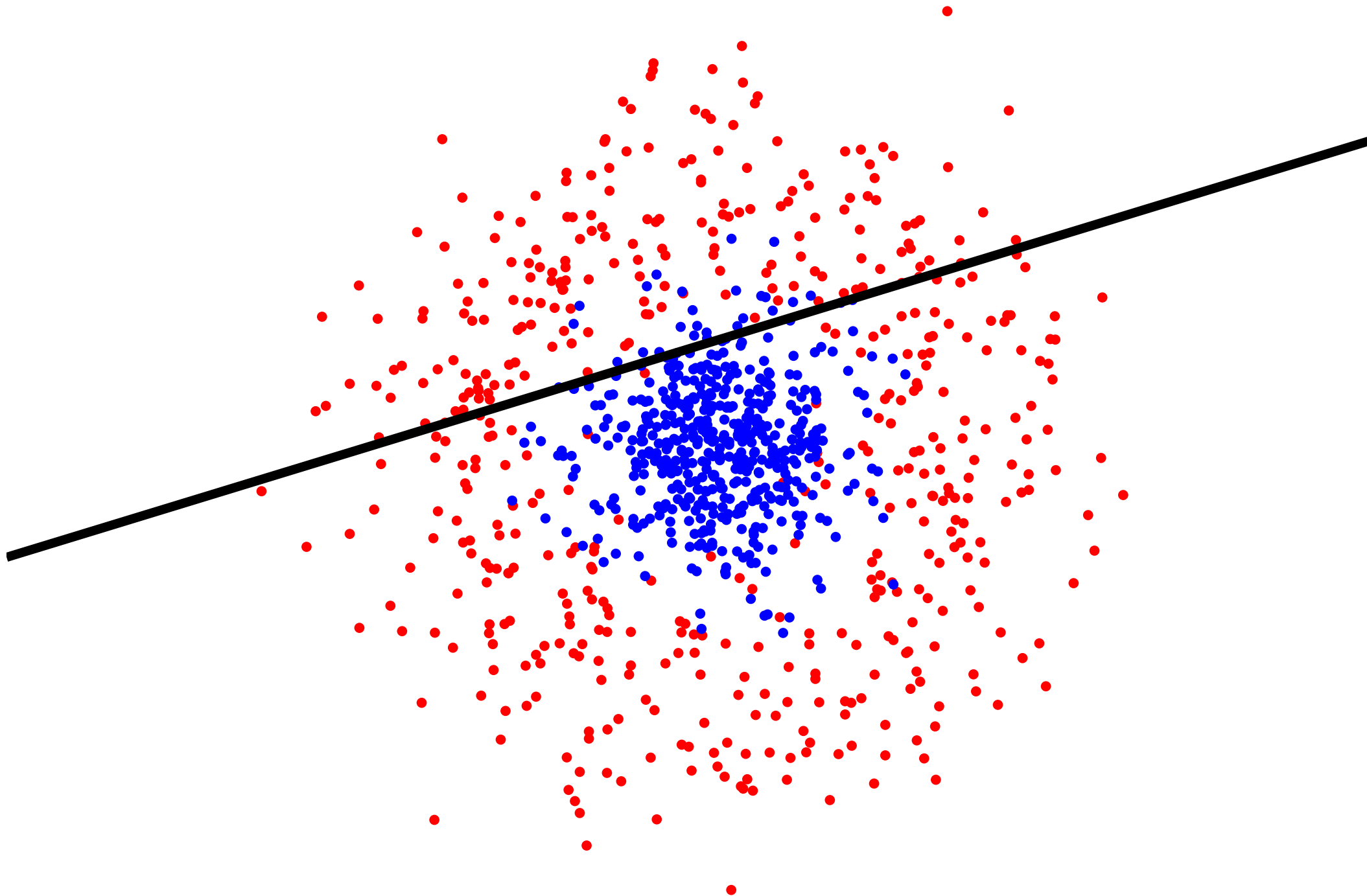


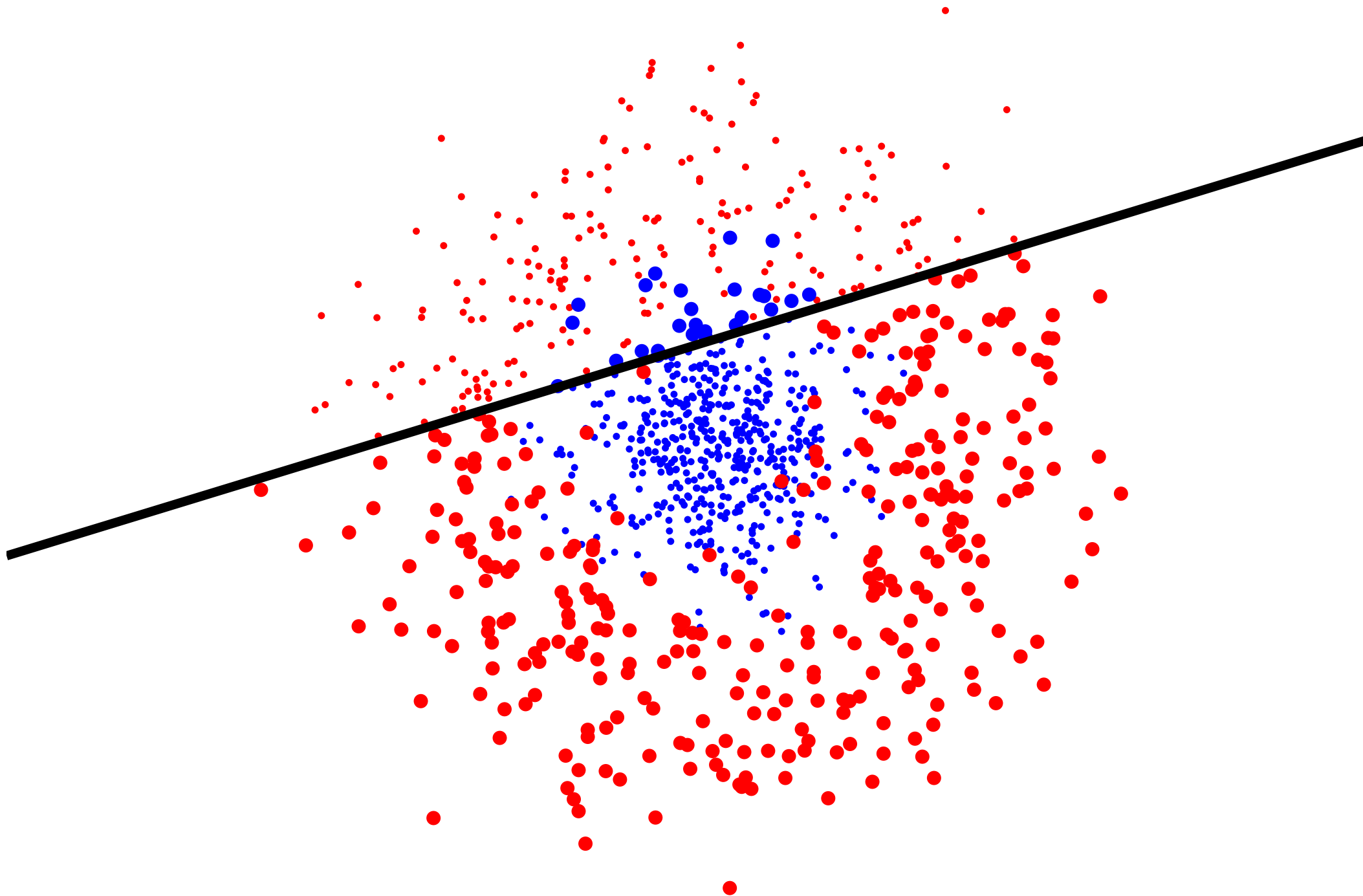


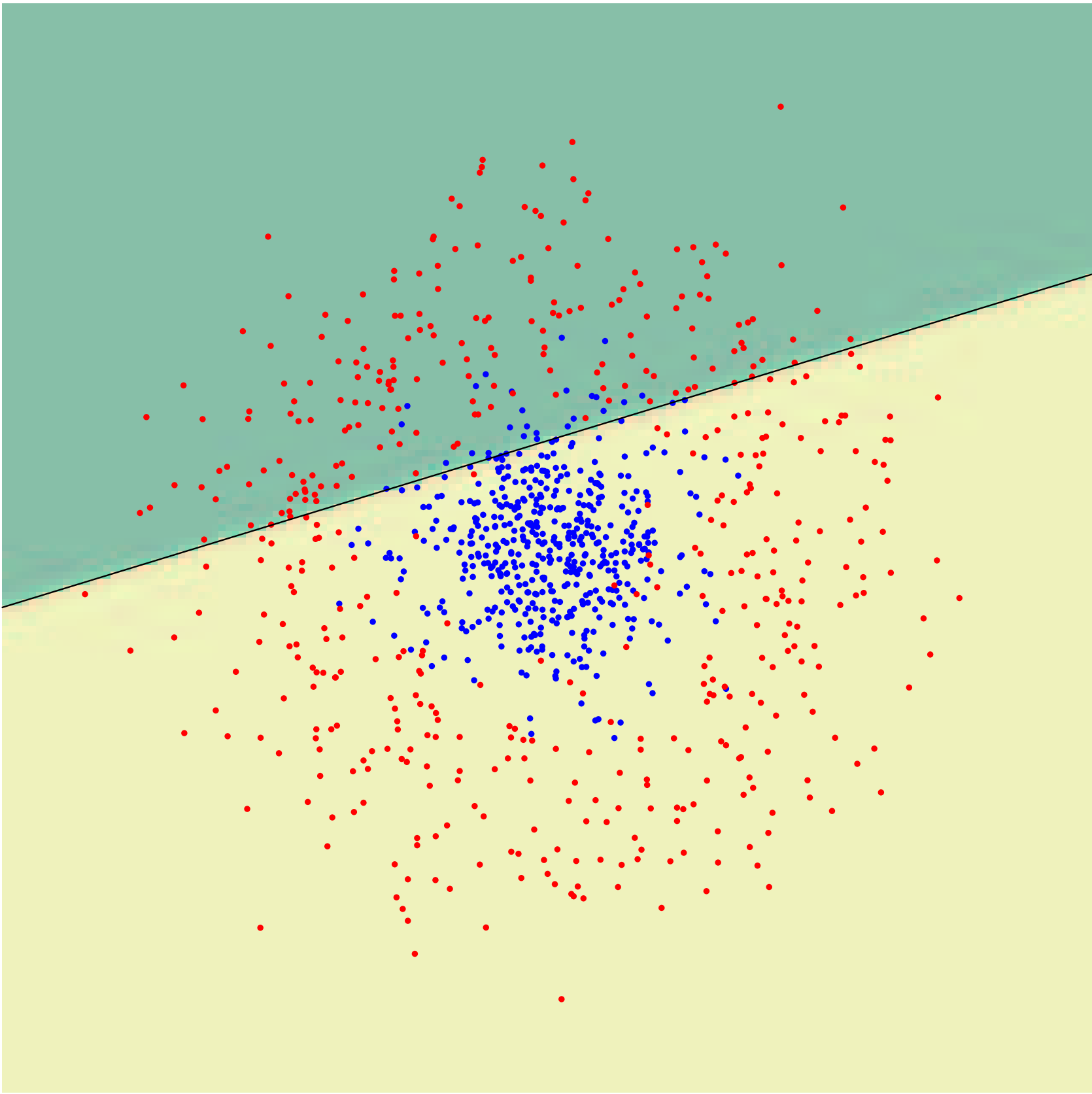


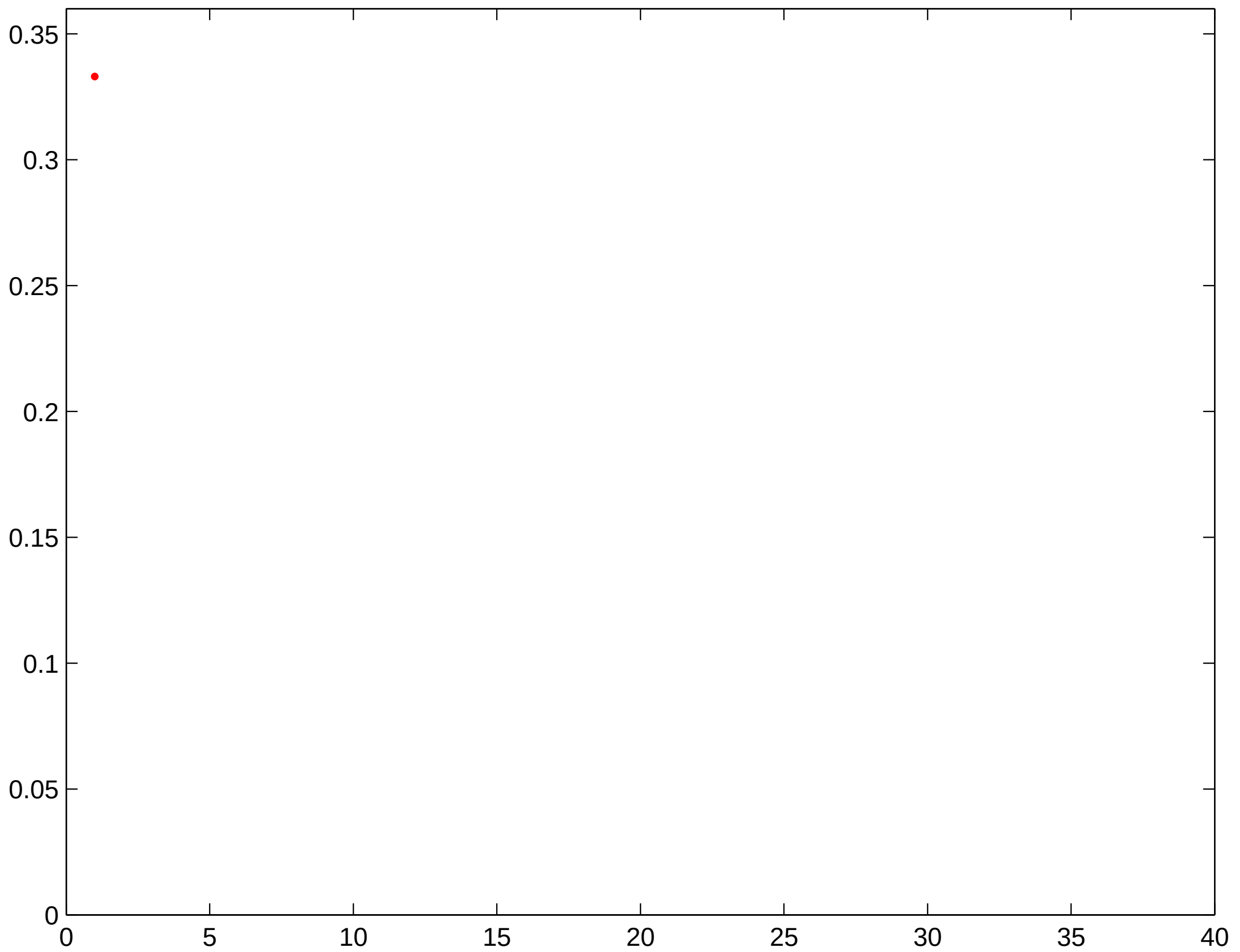


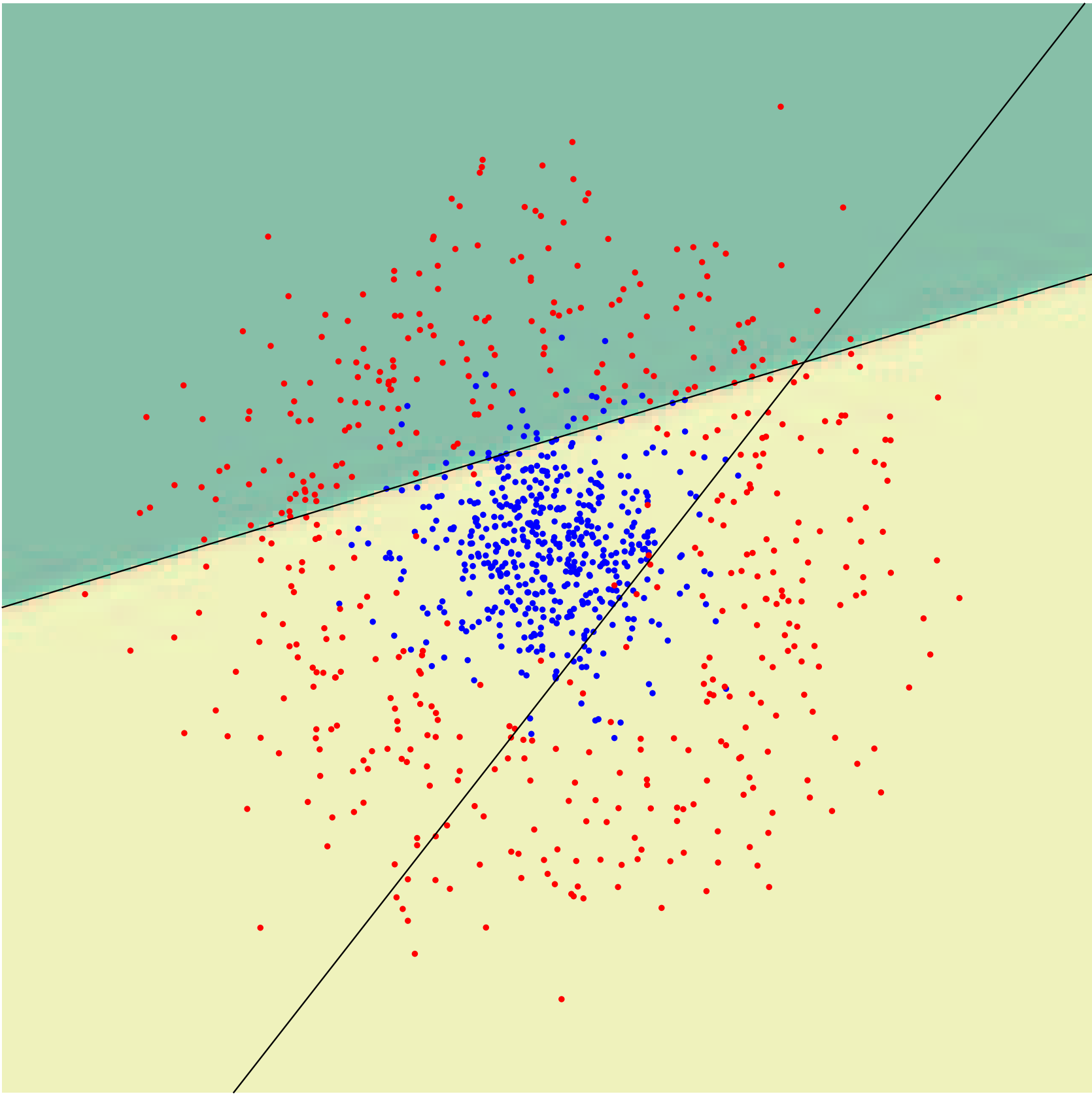


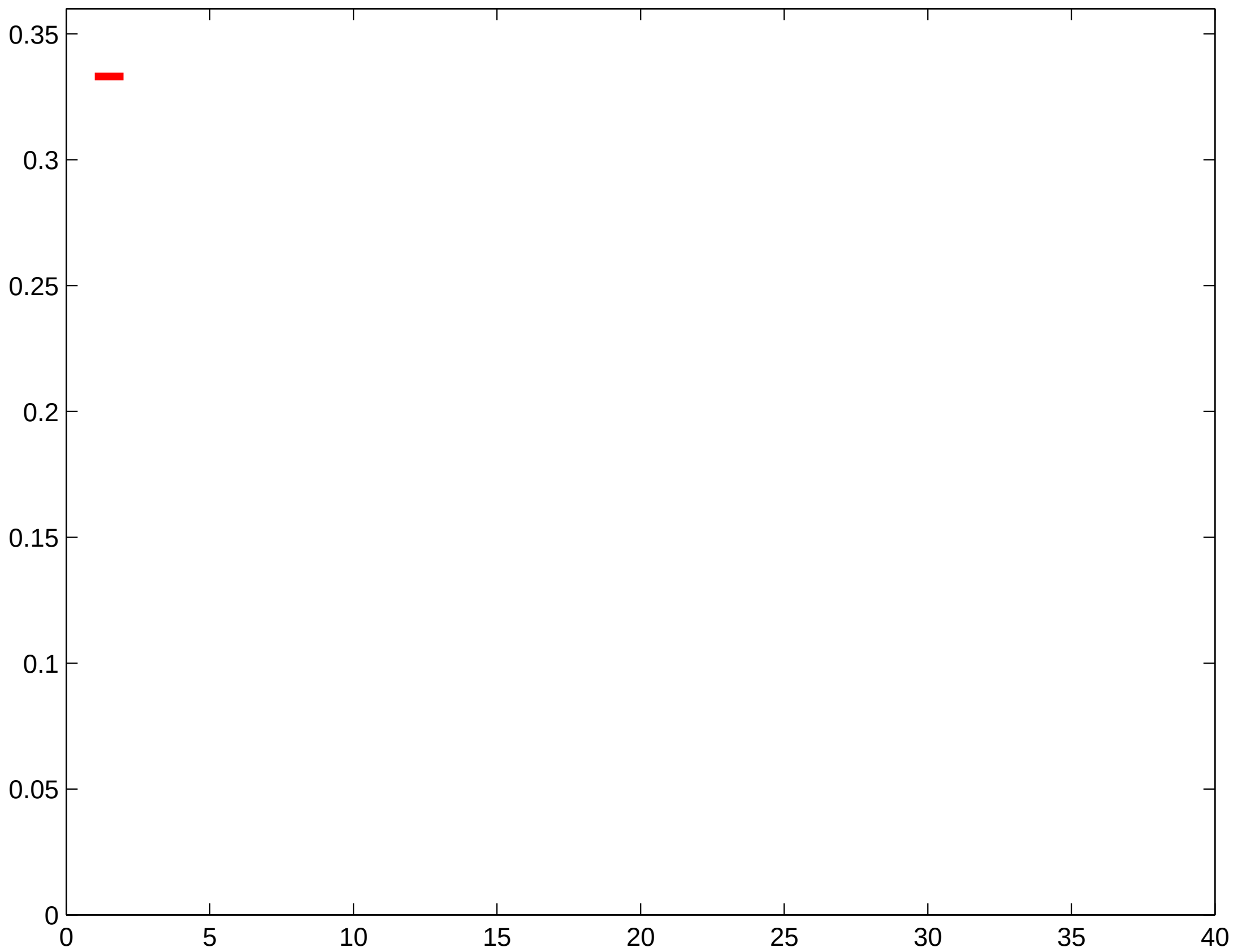


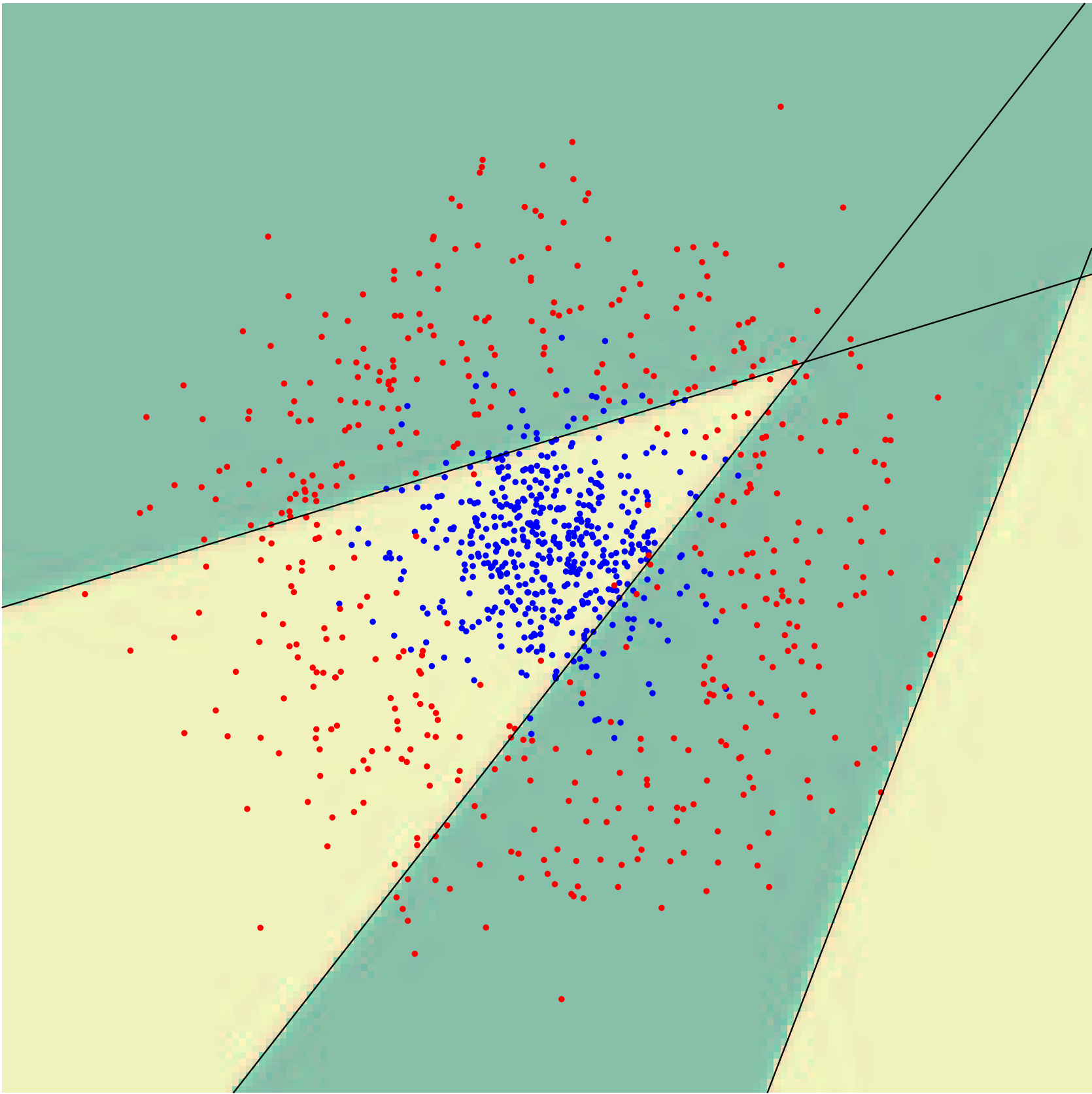


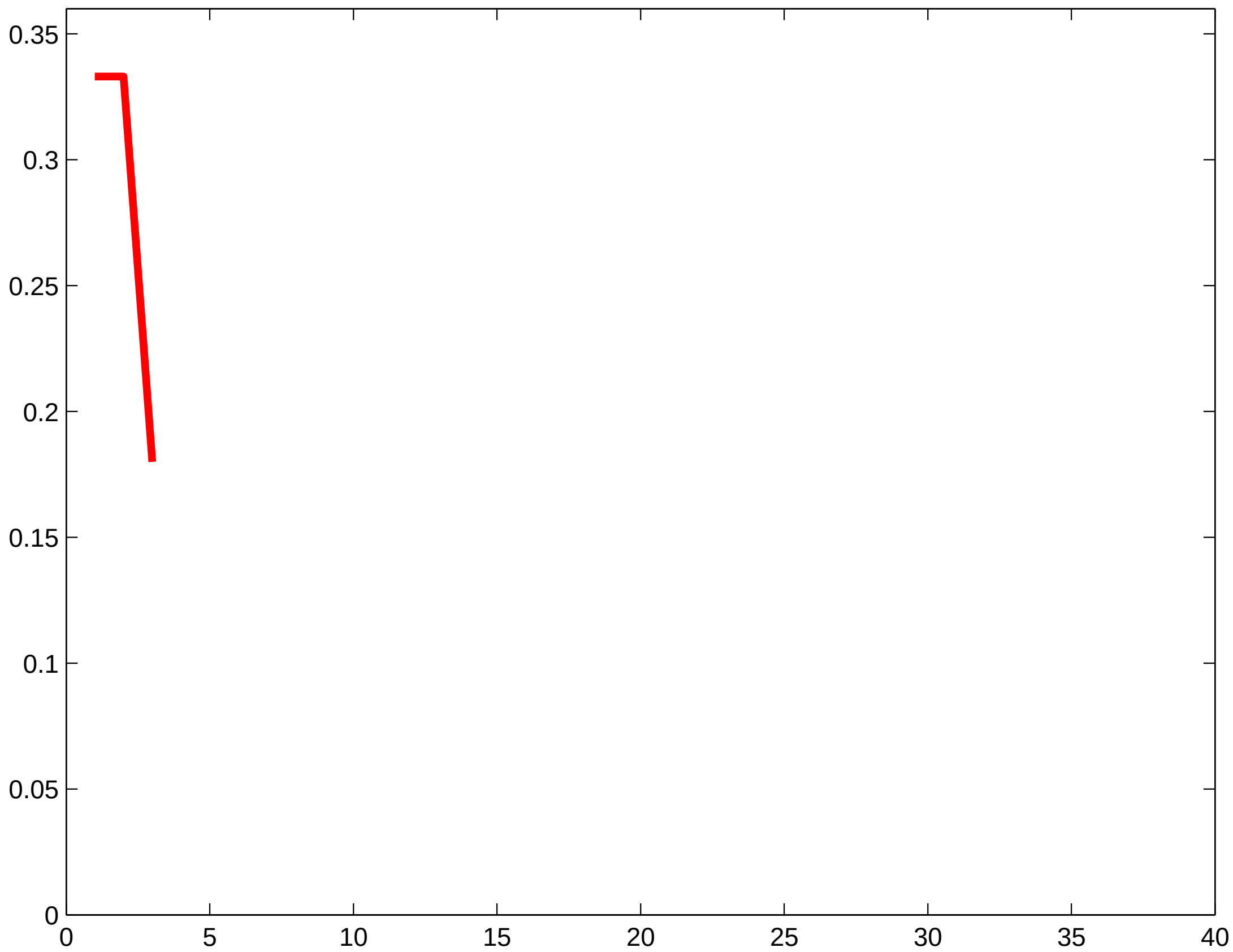


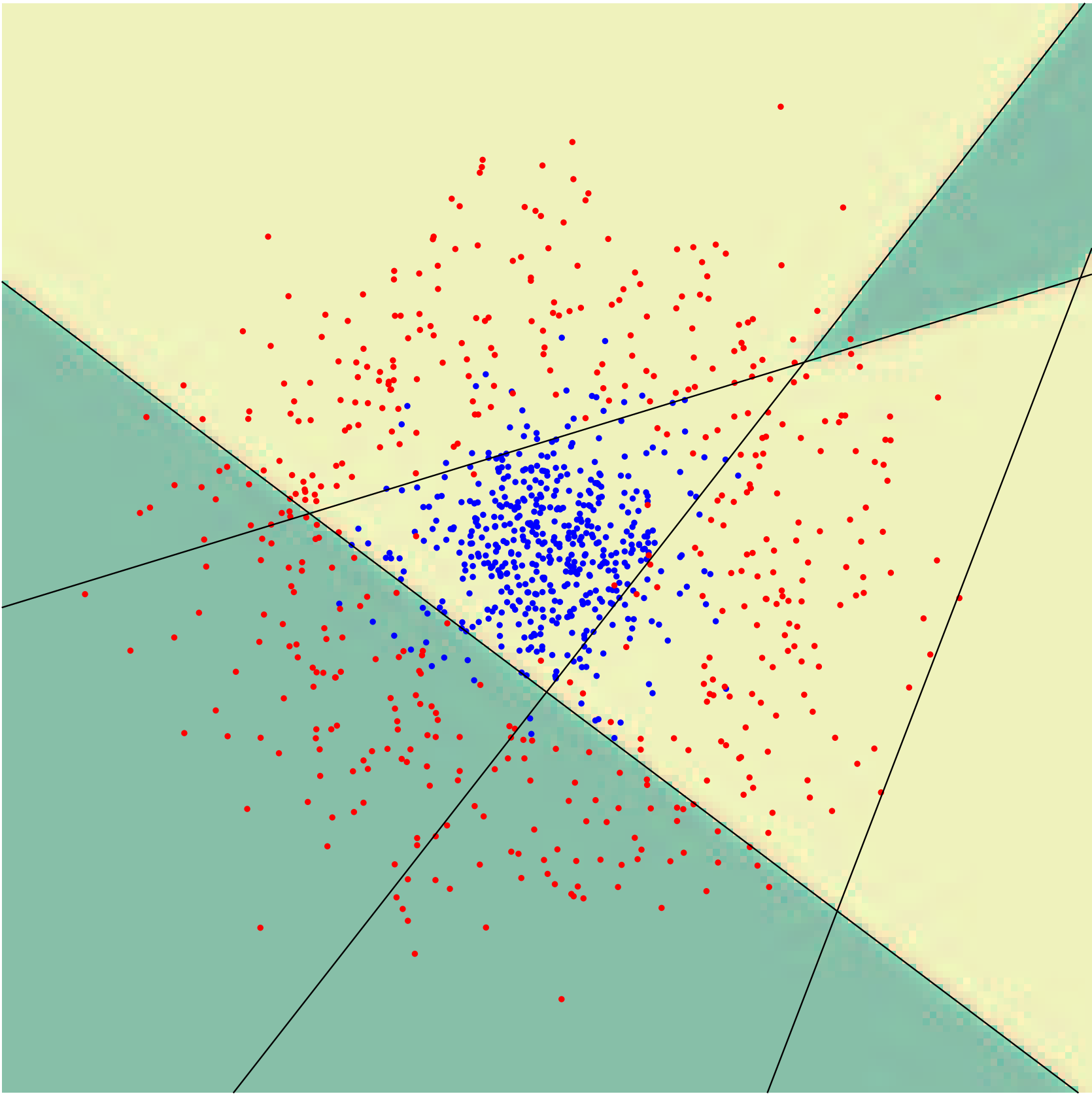


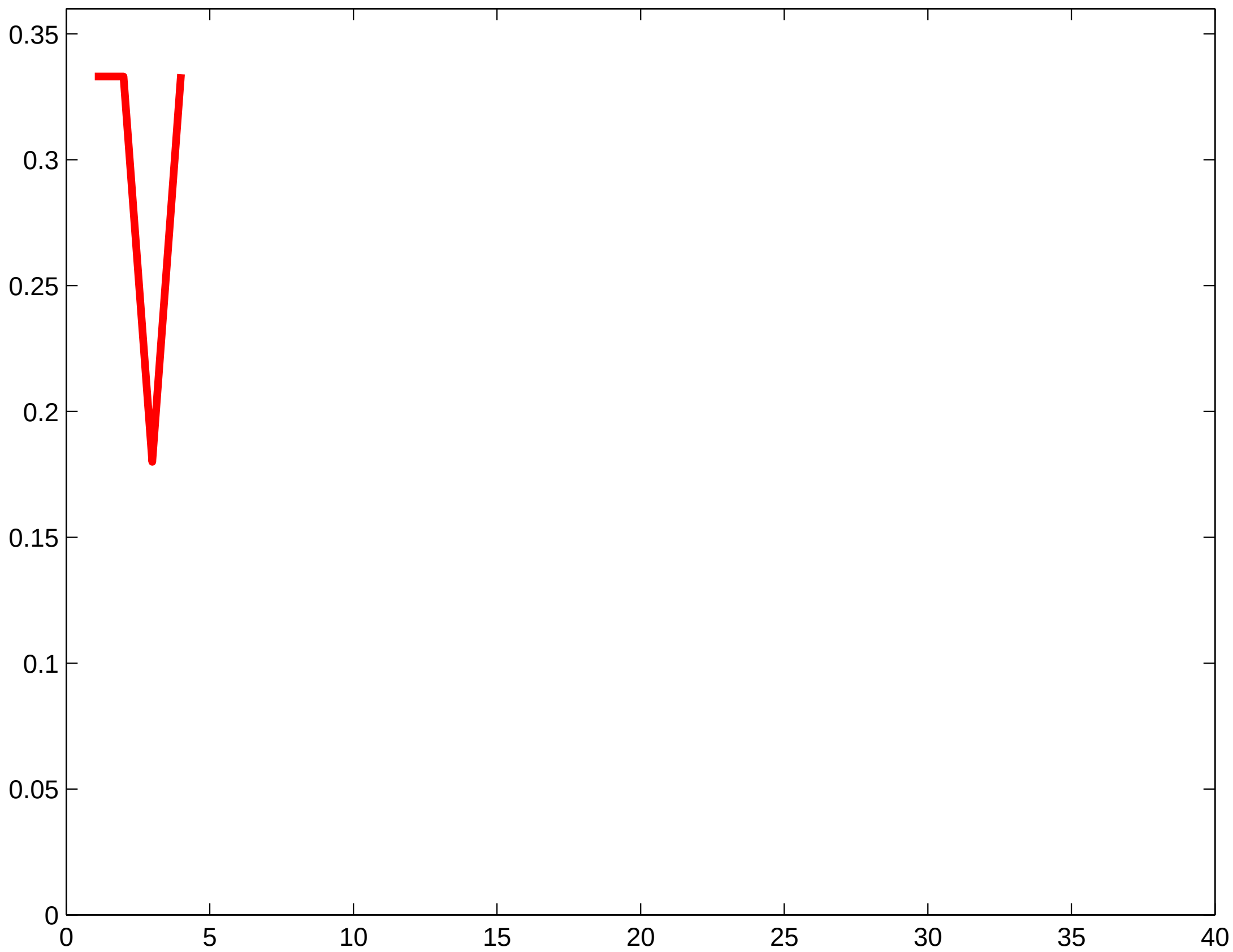


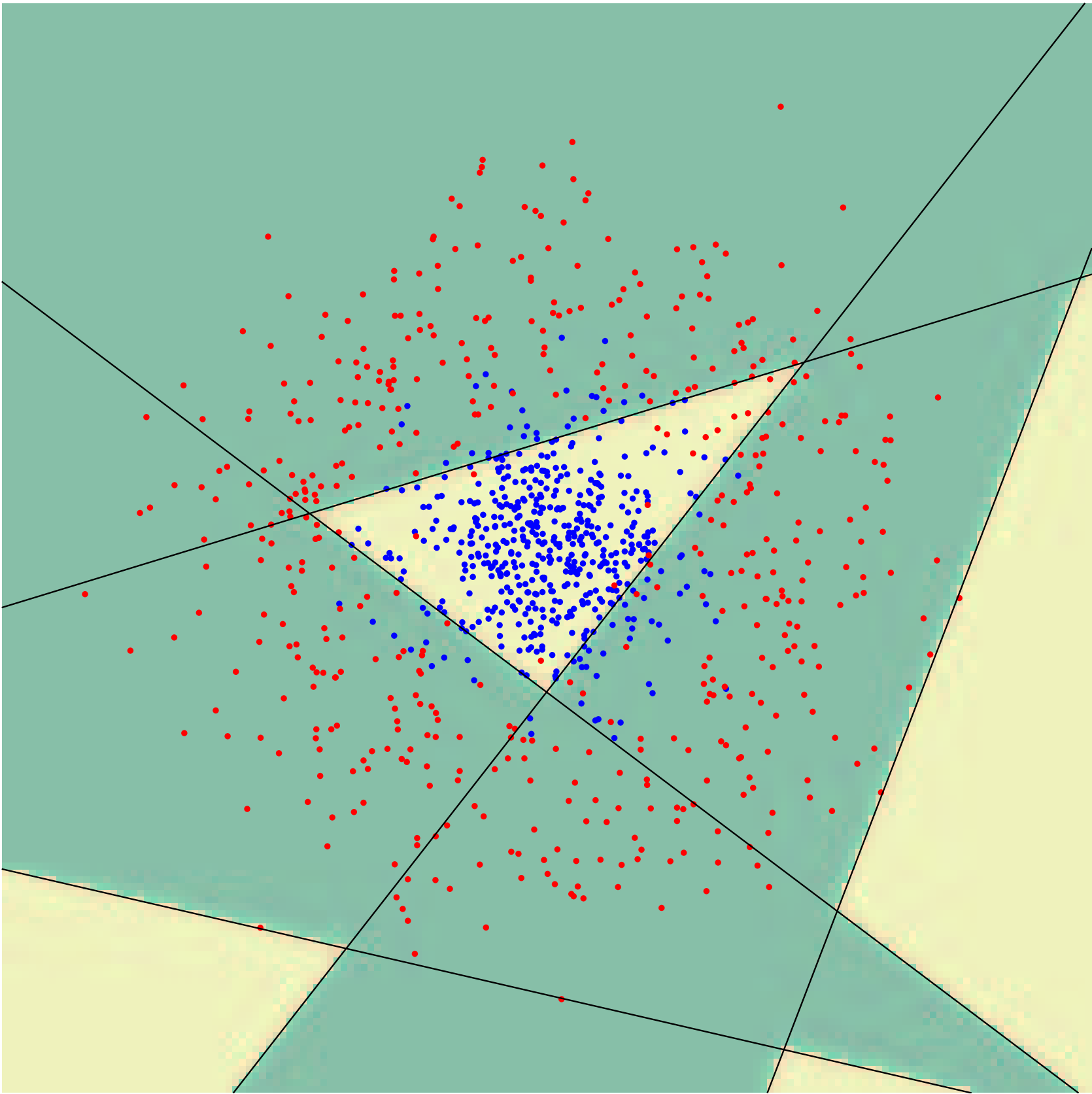


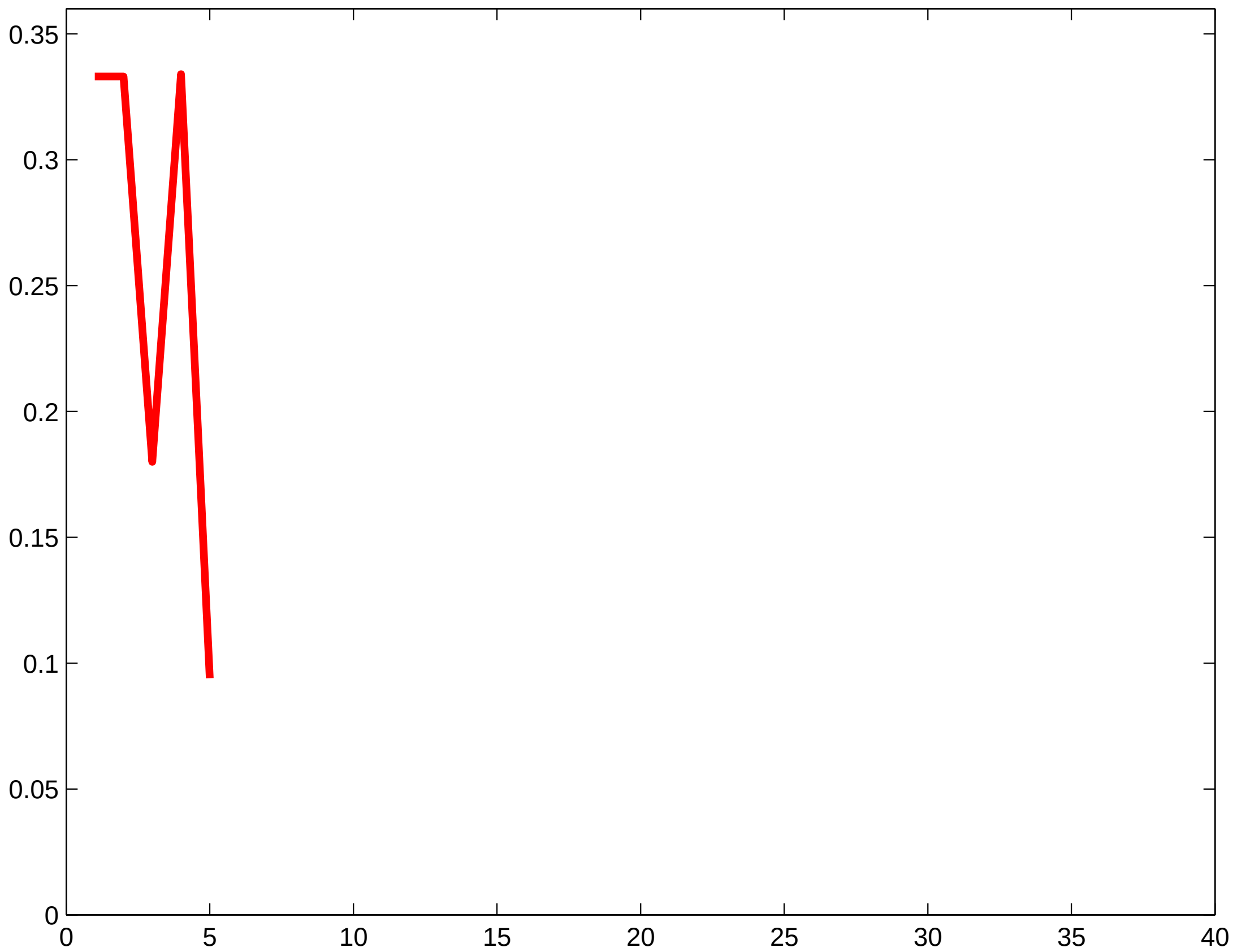


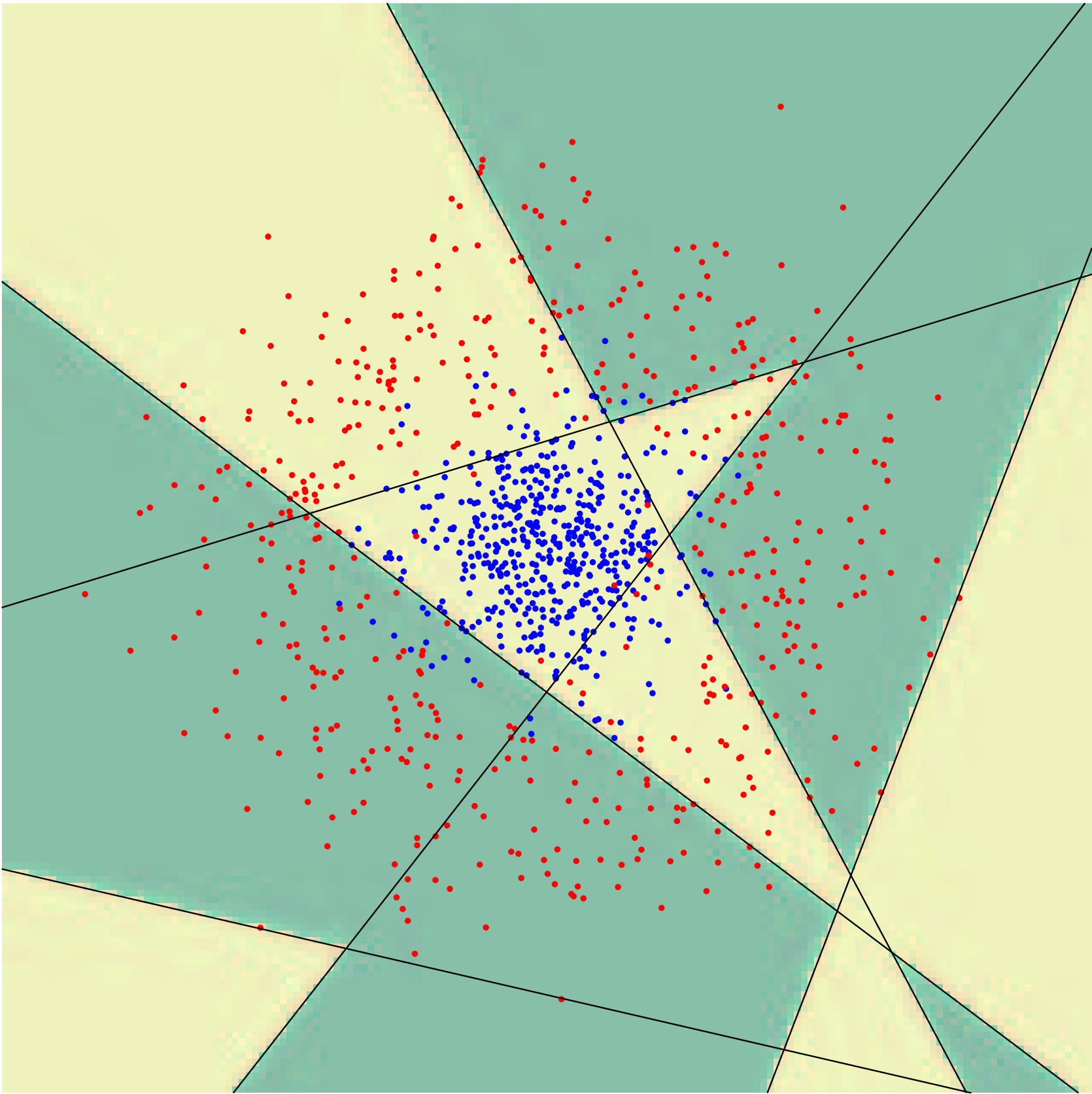


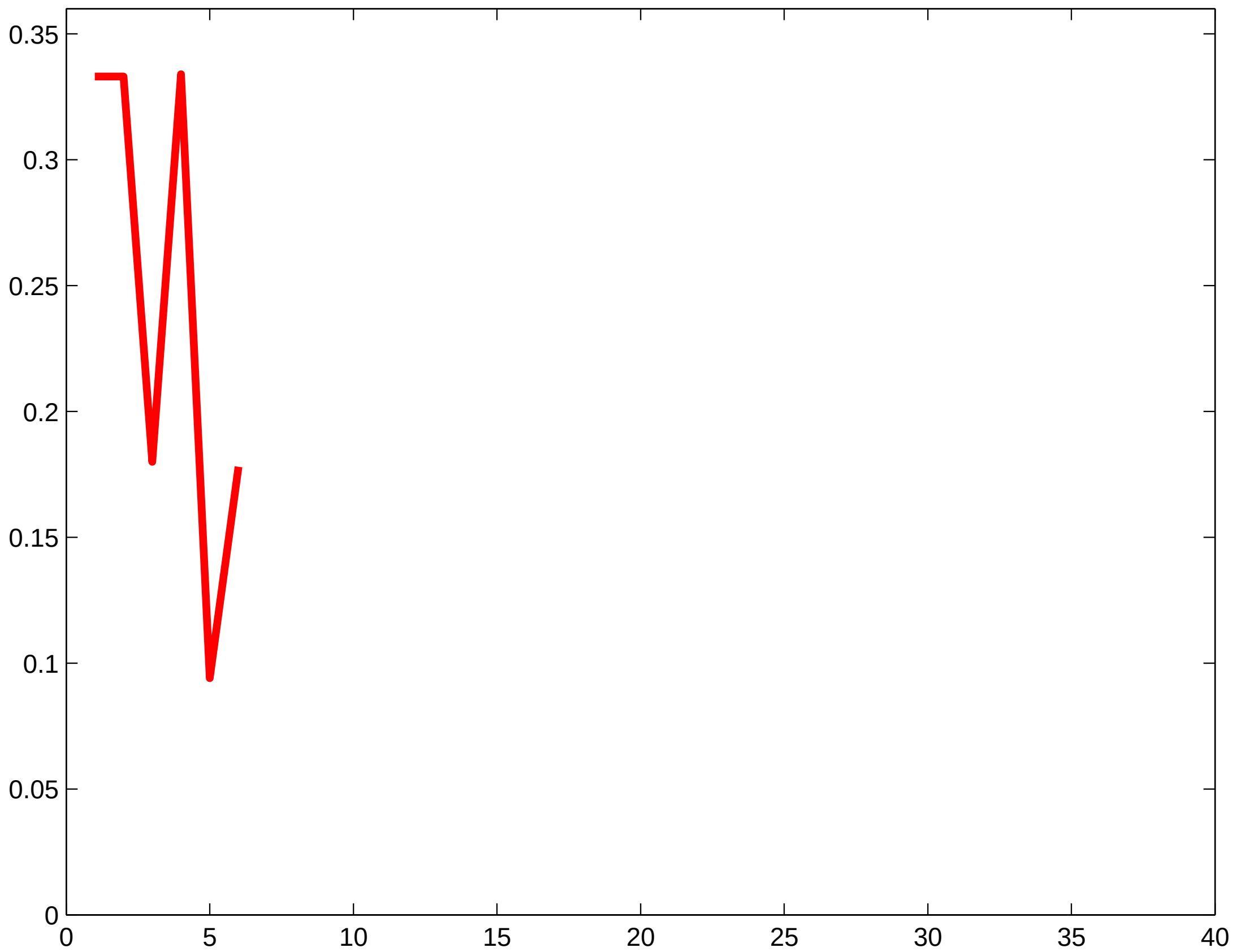


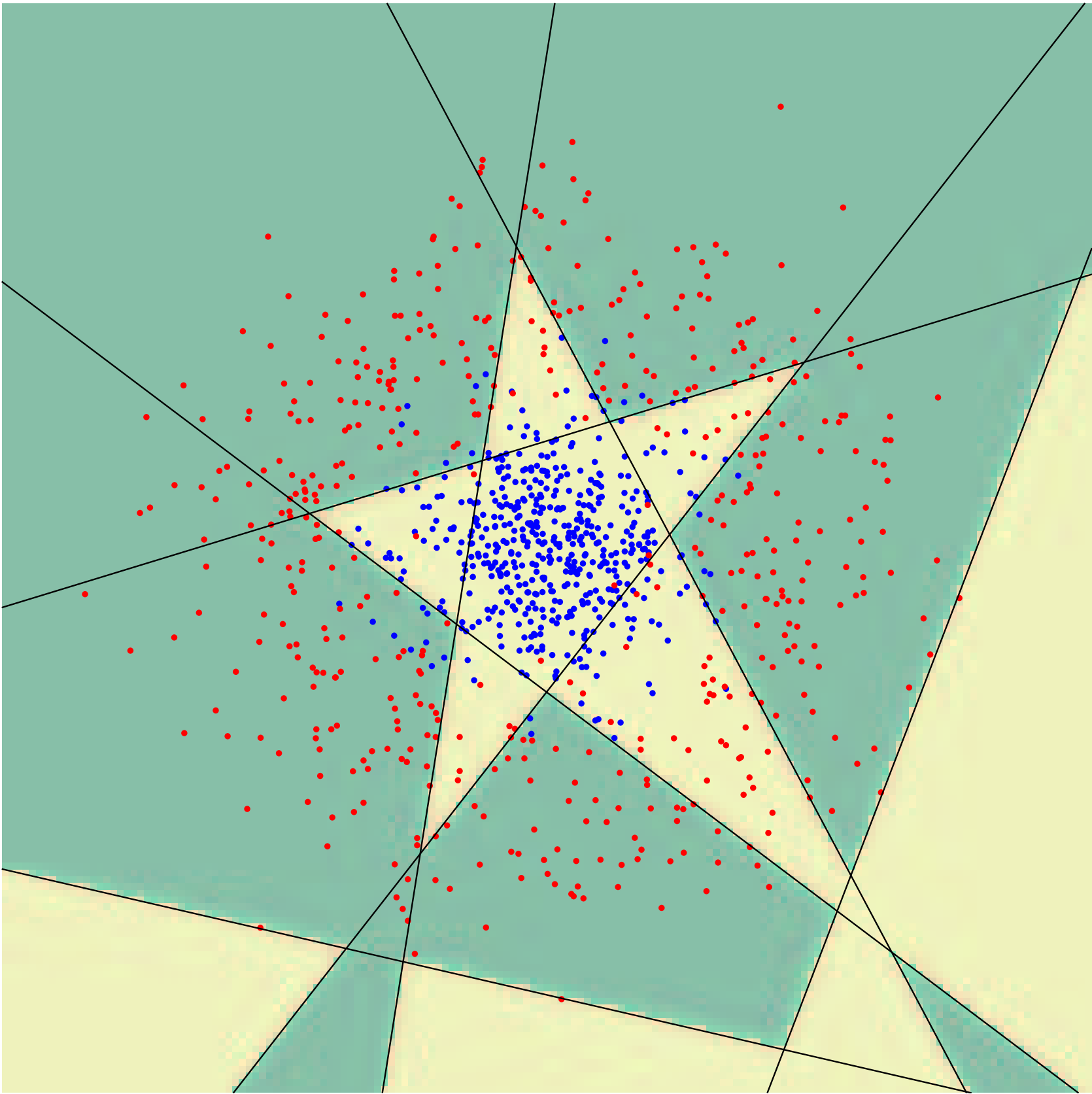


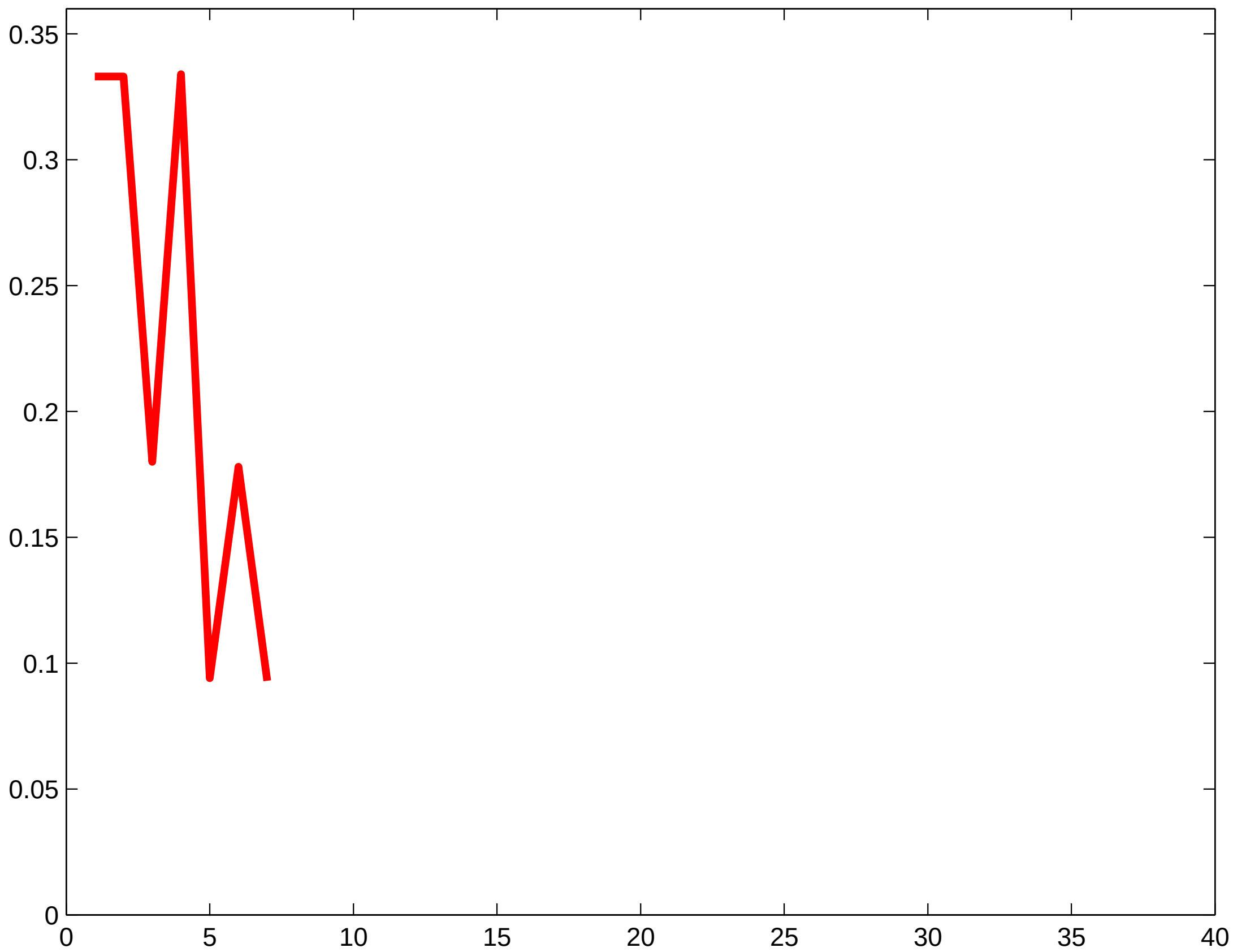


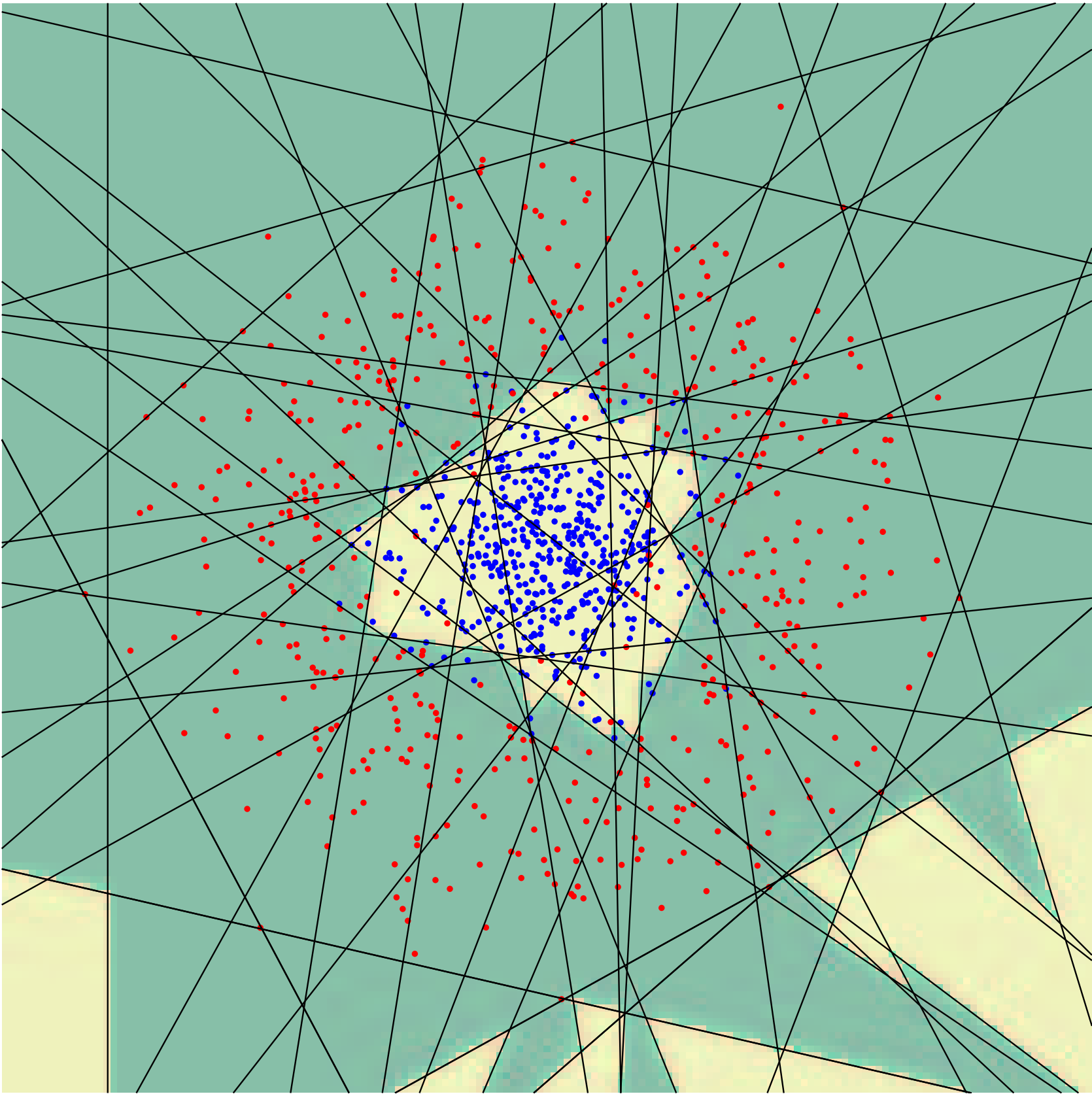


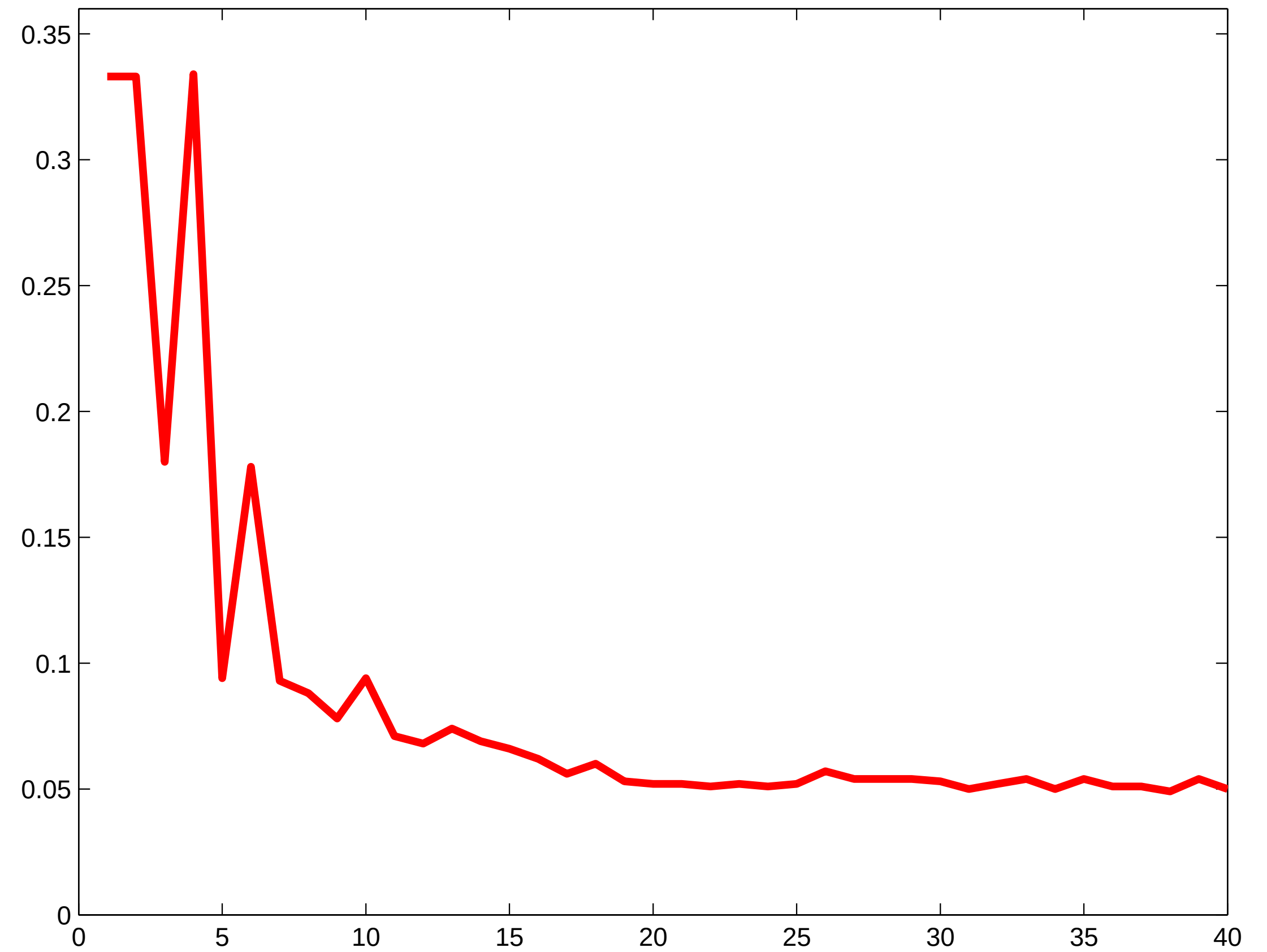




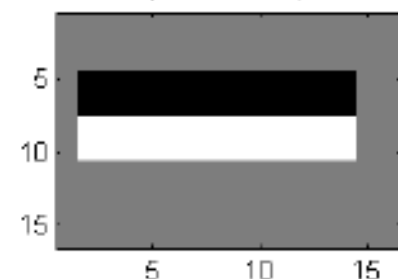




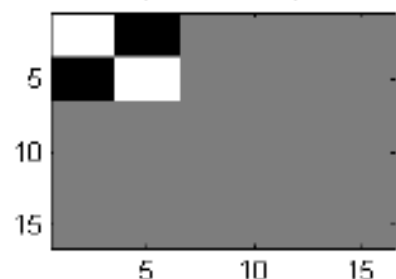




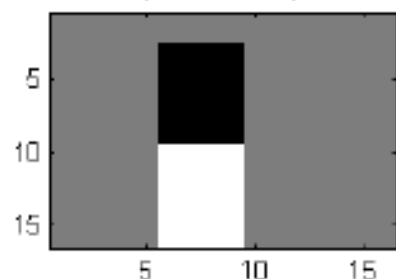
(1, 0.13, 1.93)



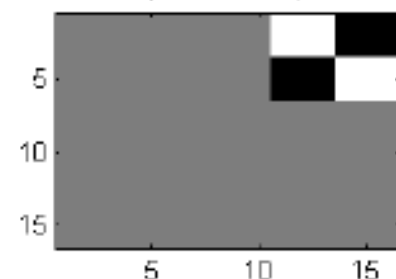
(2, 0.17, 1.59)



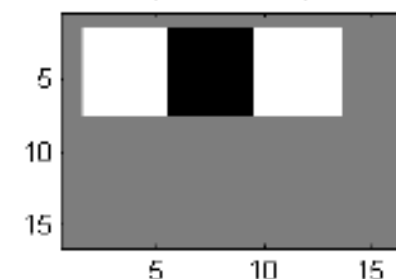
(3, 0.21, 1.30)



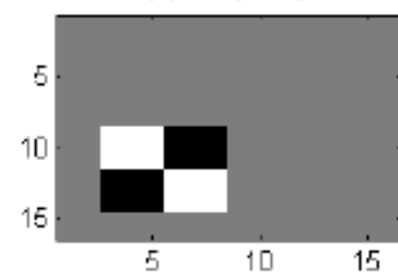
(4, 0.25, 1.12)



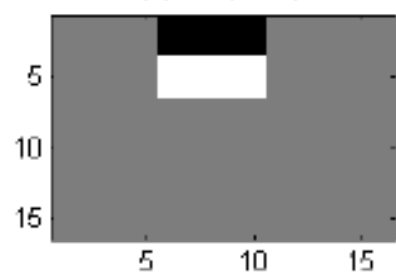
(5, 0.26, 1.05)



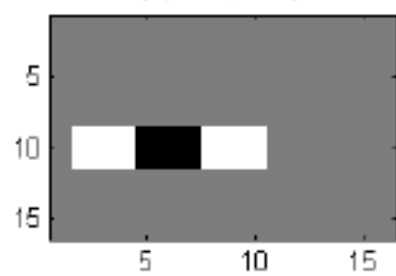
(6, 0.26, 1.04)



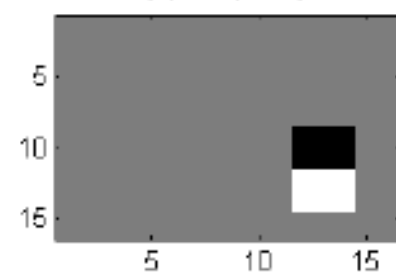
(7, 0.30, 0.82)



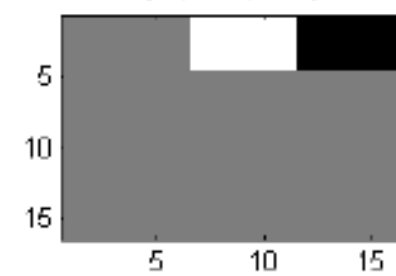
(8, 0.31, 0.79)



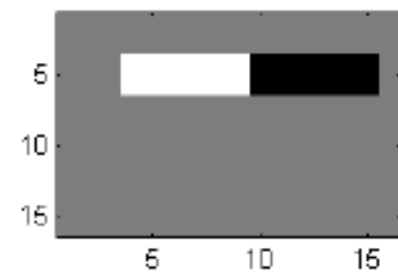
(9, 0.33, 0.72)



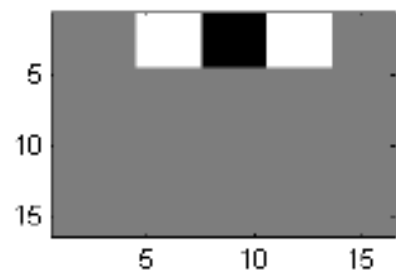
(10, 0.31, 0.81)



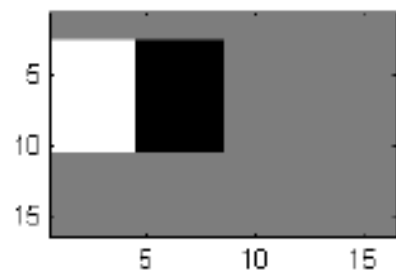
(11, 0.32, 0.78)



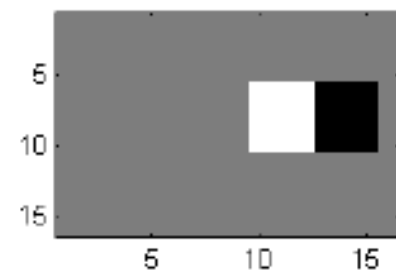
(12, 0.34, 0.65)



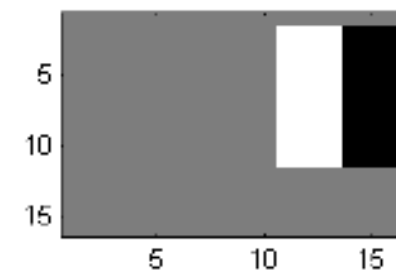
(13, 0.34, 0.64)



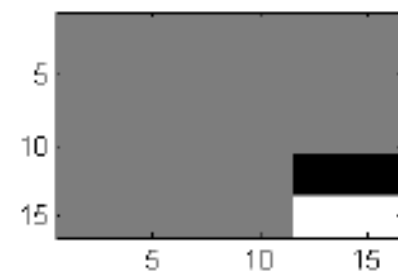
(14, 0.35, 0.60)



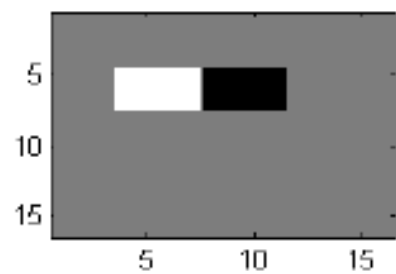
(15, 0.34, 0.67)



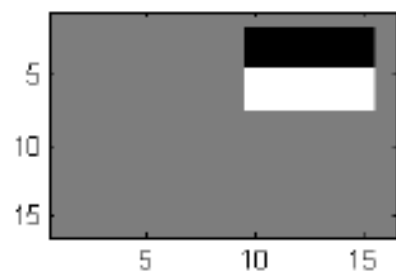
(16, 0.36, 0.59)



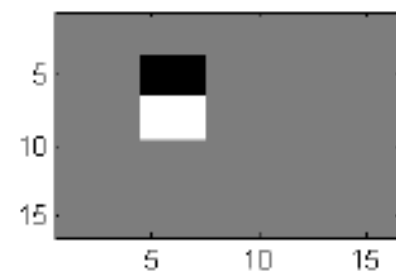
(17, 0.36, 0.57)



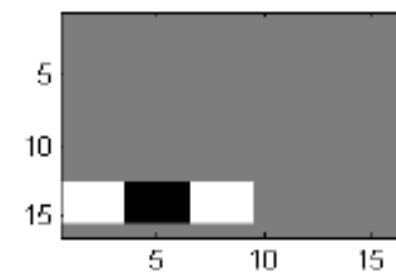
(18, 0.36, 0.56)



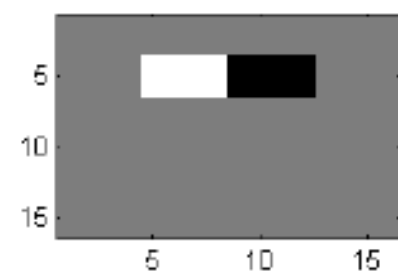
(19, 0.36, 0.57)



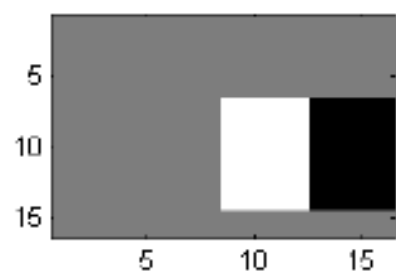
(20, 0.36, 0.56)



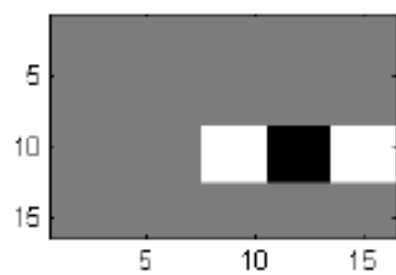
(21, 0.36, 0.56)



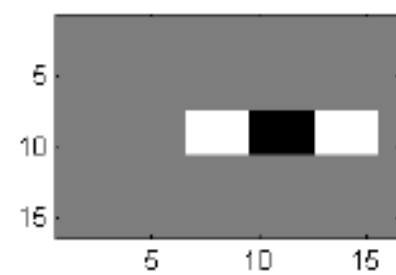
(22, 0.36, 0.55)



(23, 0.37, 0.52)



(24, 0.35, 0.61)



(25, 0.38, 0.49)

