

Learning for Time-Constrained Sequential Detection

the algorithmic core of fast (face) detection algorithms

Jiri Matas

work done with Jan Sochman, Michal Perdoch and Jan Cech

Center for Machine Perception
Czech Technical University, Prague



- **Time to decision vs. precision trade-off** is inherent in many computer vision problems, e.g. in detection, recognition and matching.
- Often the trade-off is not explicit stated in the problem formulation, but decision time clearly influences both the design and the impact of a method - time is thus implicitly considered an important characteristic of detection and recognition algorithms.

Example (face detection):

- Viola-Jones 2002: 20000+ Google Scholar citations.
- Schneiderman-Kanade 1998: 400 citations.

Schneiderman-Kanade has (had) smaller error rates, it was one of the first methods to apply machine learning, but *it is 1000x slower*.

- Wald's sequential analysis (Wald, 1947) provides a building block for quasi-optimal solutions of time-constrained recognition problems.
- It is not be clear how to measure time to decision.
Algorithmic complexity $\mathcal{O}(\cdot)$? The "hidden" constants may be huge.
Running time? Depends on the hardware.
In Wald's statistical decision theory, the number of measurements is minimized.
- We assume (and show) that the number of measurements is a good substitute for time-to-decision.
- Wald's Sequential Probability Ratio Test (SPRT) deals with the decision error v. time trade-off optimally, but:
 1. ordering of measurements is assumed to be given a priori or is irrelevant (i.i.d. measurements)
 2. unlimited number of i.i.d. measurements available (implies that zero-error decision is possible!)
 3. multidimensional pdf's are assumed to be known (reduced to estimation of a pair of 1D pdfs in the i.i.d. case)
- **WaldBoost** (Sochman and Matas, 2005) integrates feature selection and ordering provided by AdaBoost with SPRT, removing some limitations of SPRT.

We present two applications of WaldBoost (1. and 2.) and two of Wald's SPRT (3. and 4.) in computer vision algorithms:

- Application 1: **Optimal Cascaded Detector of the Viola-Jones type.**
We formalize the problem as constrained optimization, showing that Waldboost algorithm provides a quasi-optimal solution.
The approach is demonstrated on face detection (Sochman and Matas, CVPR 05).
- Application 2: **Fast Detector of Similarity-invariant Regions.** Given an executable of a carefully designed detector of repeatable "interest points" that provides an unlimited number of positive (interesting region) and negative examples, Waldboost learns an approximation of the detector whose output differs slightly from the "supervisor" and is significantly faster. (Matas and Sochman, ICRA 2007)
Might generalise to "fast approximation of a black box algorithm".
- Application 3: **Correspondence Selection with Sequential Cosegmentation** (Cech, Matas, Perdoch, PAMI 09). Wald's analysis applies directly, measurements are ordered.
- Application 4: **Optimal Randomised RANSAC**(Matas and Chum, ICCV 05). Wald's analysis applies directly, samples are i.i.d.

Basic notions:

x_1, x_2, \dots

ordered sequence of measurements

$S = \{S_1, \dots, S_t\}$

sequential strategy

where $S_i : \{x_1, \dots, x_i\} \rightarrow \{-1, +1, \#\}$

Sequential strategy S is characterised by:

1. α_S and β_S

errors of the first and the second kind

2. $\bar{T}_S = E_X(T_S(x))$

average evaluation time

where $T_S(x) = \arg \min_i (S_i(x) \neq \#)$

time-to-decision

Problem formulation:

$$S^* = \arg \min_S \bar{T}_S \quad \text{s.t.} \quad \beta_S \leq \beta, \\ \alpha_S \leq \alpha$$

for specified α and β

Note: The formulation above assumes $P(k), k \in \{-1, 1\}$ exist and are known.

A formulation, requiring only $P(x|k)$, where $\bar{T}_S = \max_k E_X(T_S(x|k))$ is the maximum of the expected decision times for classes $\{1, -1\}$, leads to identical results.

Sequential Probability Ratio Test (SPRT)

SPRT is a sequential strategy S^*

$$S_t^* = \begin{cases} +1, & R_t \leq B \\ -1, & R_t \geq A \\ \#, & B < R_t < A \end{cases} \quad R_t = \frac{p(x_1, \dots, x_t | y = -1)}{p(x_1, \dots, x_t | y = +1)}$$

Practical (and nearly optimal) setting for the thresholds A and B

$$A = \frac{1 - \beta}{\alpha}, \quad B = \frac{\beta}{1 - \alpha}$$

Difficulties

SPRT is an **optimal sequential test** for the formulated problem **but**:

1. if measurements not i.i.d then their order must be given.
2. multidimensional pdf's $p(x_1, \dots, x_t | y = \text{class})$ have to be computed in non i.i.d case.



AdaBoost

- Selects measurements h_t and combines them into classifier
- The output of the AdaBoost algorithm is a classification function

$$H_T(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

- Classification is given by $\text{sign}(H_T(x))$

Measurement selection and ordering

⇒ The outputs of selected weak classifiers are taken as measurements used in SPRT.

h_1, h_2, \dots ordered sequence of measurements

Wald's SPRT requires the knowledge (an estimate) of the likelihood ratio

$$\hat{R}_t(x) = \frac{p(h^{(1)}(x), \dots, h^{(t)}(x) | y = -1)}{p(h^{(1)}(x), \dots, h^{(t)}(x) | y = +1)}$$

The weak classifiers cannot be treated as statistically independent

⇒ difficult multi-dimensional density estimation.

WaldBoost Approach.

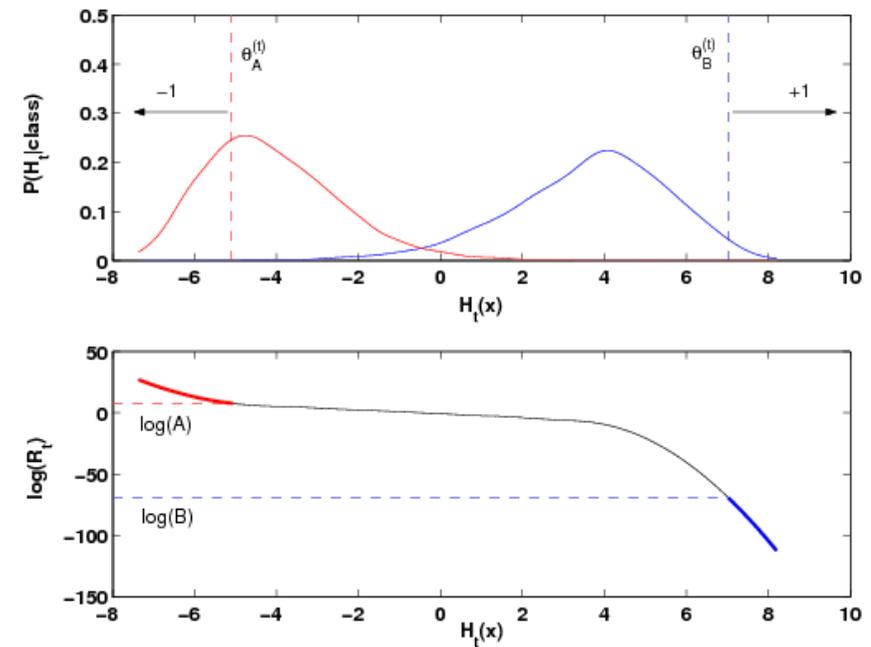
The global minimiser of AdaBoost learning is

$$\tilde{H}(x) = -\frac{1}{2} \log R(x) + \frac{1}{2} \log \frac{P(+1)}{P(-1)}.$$

A one-to-one mapping exists between $\tilde{H}(x)$ and $R(x)$

⇒ the ratio is estimated by

$$\hat{R}_t(x) = \frac{p(H_t(x) | y = -1)}{p(H_t(x) | y = +1)}.$$



Input: $(x_1, y_1), \dots, (x_l, y_l)$; $x_i \in \mathcal{X}, y_i \in \{-1, 1\}$,
desired final false negative rate α and false positive rate β .

Set $A = (1 - \beta)/\alpha$ and $B = \beta/(1 - \alpha)$

For $t = 1, \dots, T$

1. Choose best h_t as in AdaBoost
2. Estimate the likelihood ratio R_t
3. Find thresholds $\theta_A^{(t)}$ and $\theta_B^{(t)}$
4. Throw away samples from training set for which
 $H_t \geq \theta_B^{(t)}$ or $H_t \leq \theta_A^{(t)}$
5. (Optional.) Bootstrap: Sample new data into the training set

end

Output: strong classifier H_T and thresholds $\theta_A^{(t)}$ and $\theta_B^{(t)}$.

Given: $H_T, \theta_A^{(t)}, \theta_B^{(t)}, \gamma$

Input: a classified object x

For $t = 1, \dots, T$ (*SPRT execution*)

If $H_t(x) \geq \theta_B^{(t)}$, classify x to the class +1 and terminate

If $H_t(x) \leq \theta_A^{(t)}$, classify x to the class -1 and terminate

end

If $H_T(x) > \gamma$, classify x as +1. Classify x as -1 otherwise.

Optionally, if the final stage is reached very rarely, make the final decision by a classifier of arbitrary complexity, using $h_t(x)$ or in general x .

Face detection specific properties

- Highly unbalanced face and background class sizes and complexities.
- Missed detection more serious than false positive.

WaldBoost learning for face detection

We therefore set $\beta = 0$ in WaldBoost. Thus

$$A = \frac{1 - 0}{\alpha} = \frac{1}{\alpha}, \quad B = \frac{0}{1 - \alpha} = 0$$

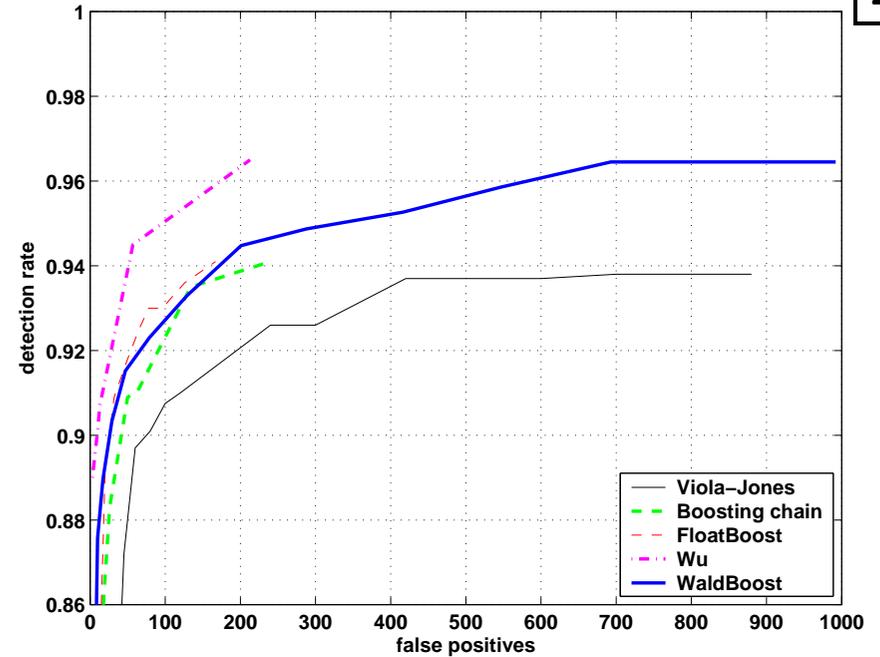
and the SPRT strategy becomes

$$S_t^* = \begin{cases} +1, & R_t \leq 0 \\ -1, & R_t \geq 1/\alpha \\ \#, & 0 < R_t < 1/\alpha \end{cases}$$

⇒ Since $R_t > 0$, only allowed decision is the classification to the background class

⇒ Only background class pruned during training

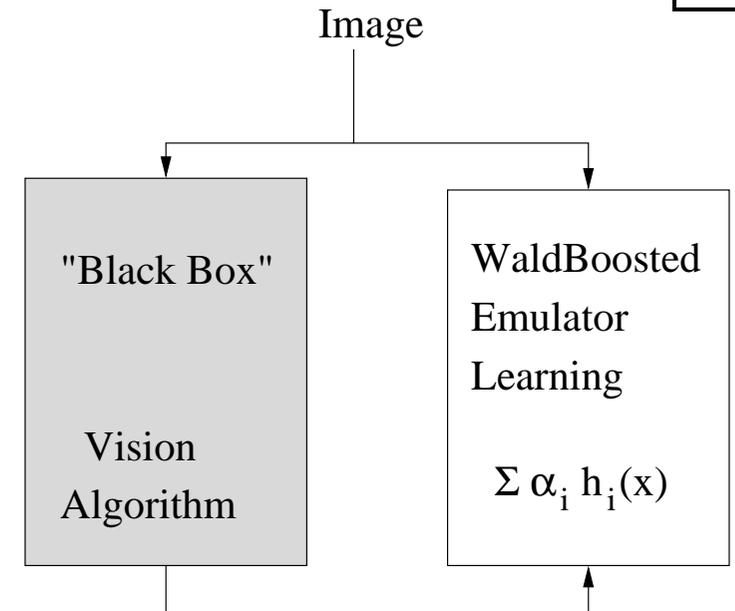
Face Detection Results



Method	WB	VJ	FloatBoost	Boosting chain	Wu
#wc	400	4297	2546	700	756
\bar{T}_S	10.84	8	18.9	18.1	N/A

- WaldBoost results in the fastest classifier (see \bar{T}_S) among those with comparable detection rate.
- Least number of weak classifiers needed in WaldBoost.
- Detection rates are comparable to the state-of-the-art methods.

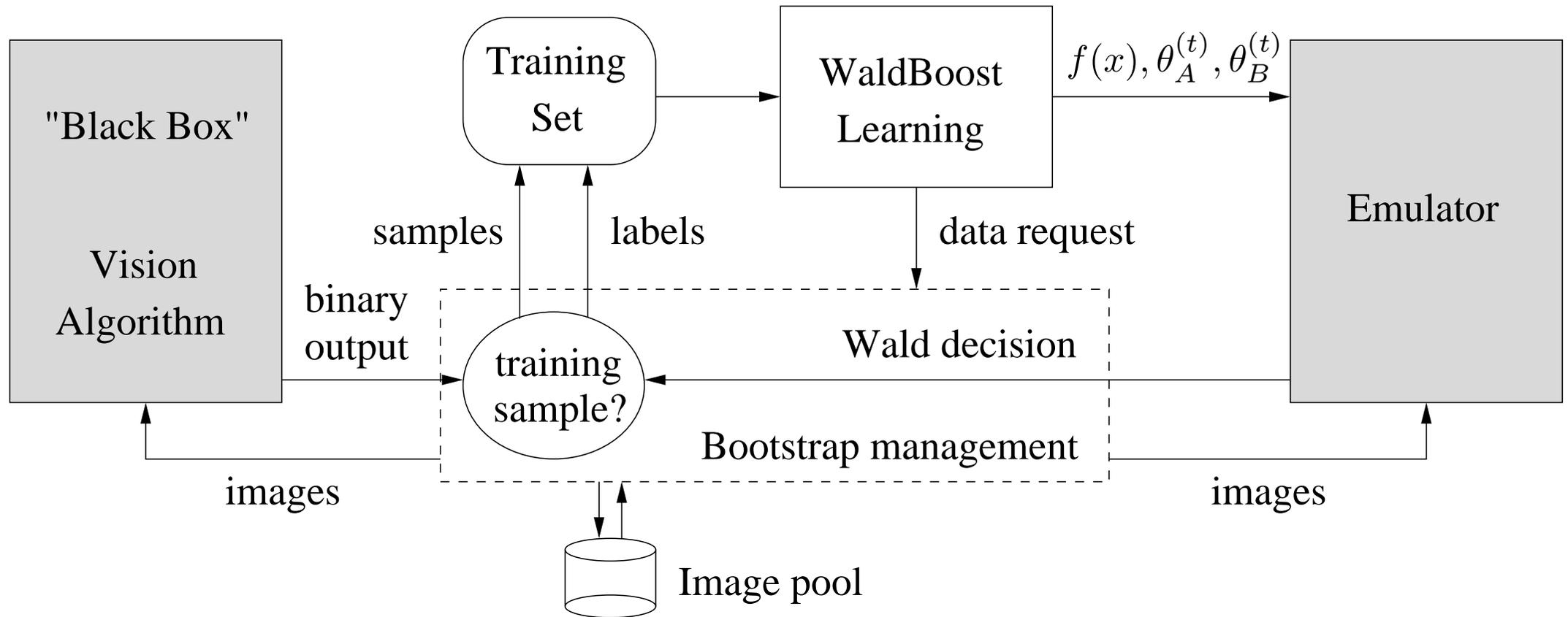
- Let's assume an executable of a binary-valued (and slow) vision algorithm and suitable set of efficient weak classifiers \mathcal{H} "suitable" for the algorithm is given.
- WaldBoost is trained on a set of examples generated by the "black box". Optimizing time to decision given limits on false positive and false negative rates
- Result: fast version of the black box algorithm



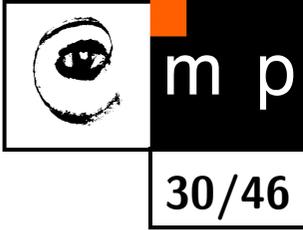
Notes:

- If the emulator is not fast enough, search for better \mathcal{H} .
This is very different from spending man-months optimizing and/or finding fast approximation of a process (e.g. SIFT v. SURF)!
- Your own MATLAB implementation can serve as the "black box".
- Through false positives and negatives, the most complicated (from the computational point of view) examples are dropped.

Applying WaldBoost 2: Emulating a Black Box Algorithm



Applying WaldBoost 2: Emulating Kadir and Hessian-Laplace Interest Point Detectors

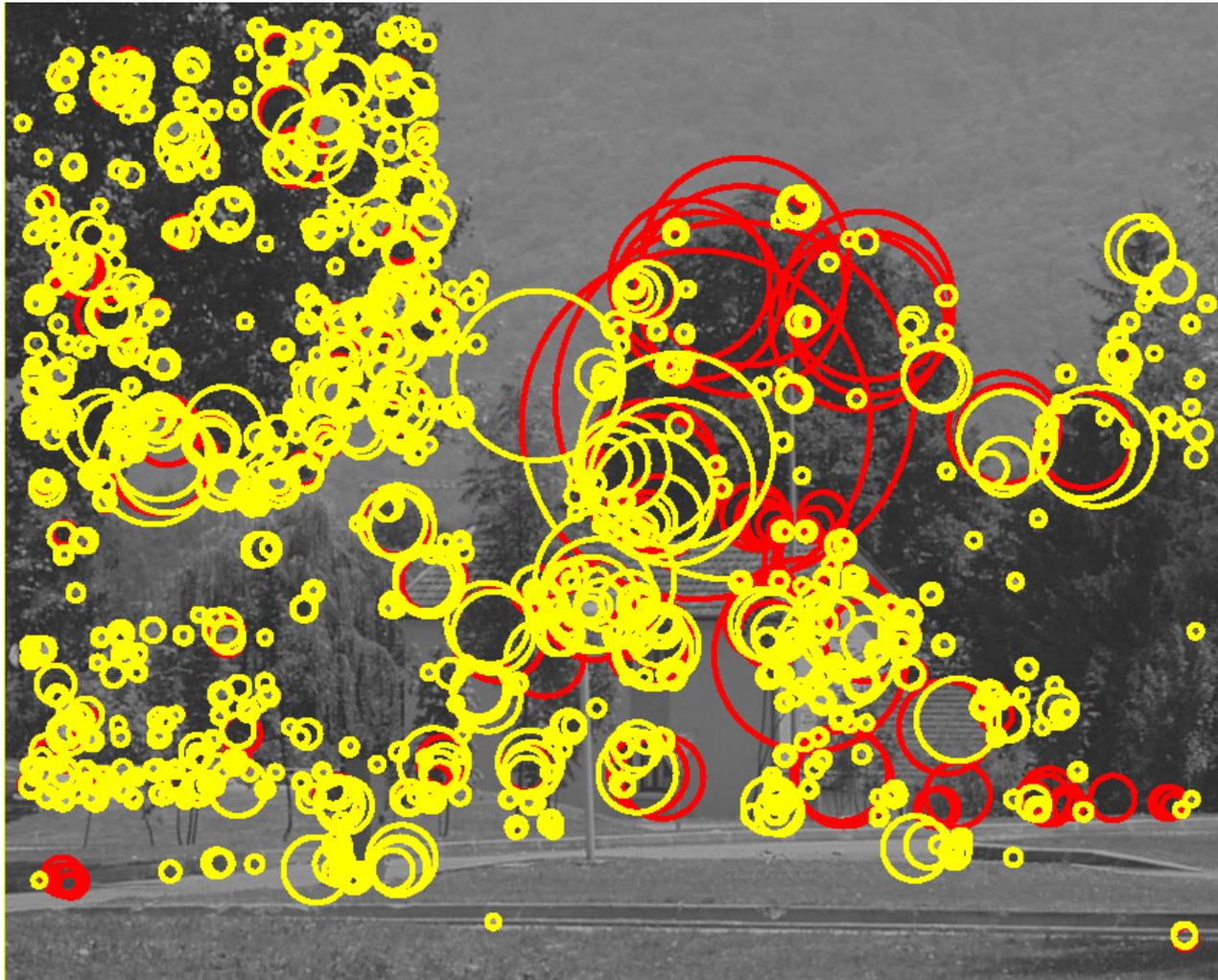


Motivation:

- Executables of both methods available at `robots.ox.ac.uk`
- Both detectors are scale-invariant (not affine), which is easily implemented via a scanning window + a sequential test.
- Standard testing protocol exists [Mikolajczyk et al. IJCV05]
- Kadir's detector very slow (100x slower than Difference of Gaussians).
- Hessian-Laplace is close to state of the art.

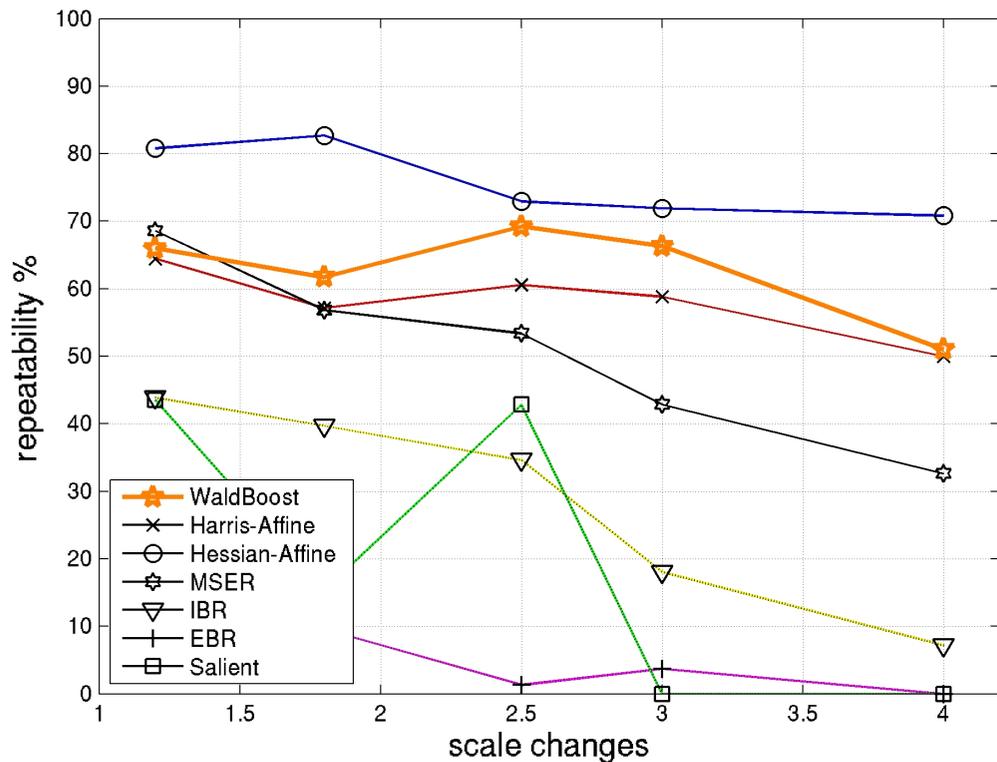
Implementation:

- Choice of \mathcal{H} : various filters computable with integral images of intensity and intensity squared: difference of rectangular regions, variance in a window.
- False negative rate $\alpha = 0.2$, false negative rate $\beta = 0.2$.

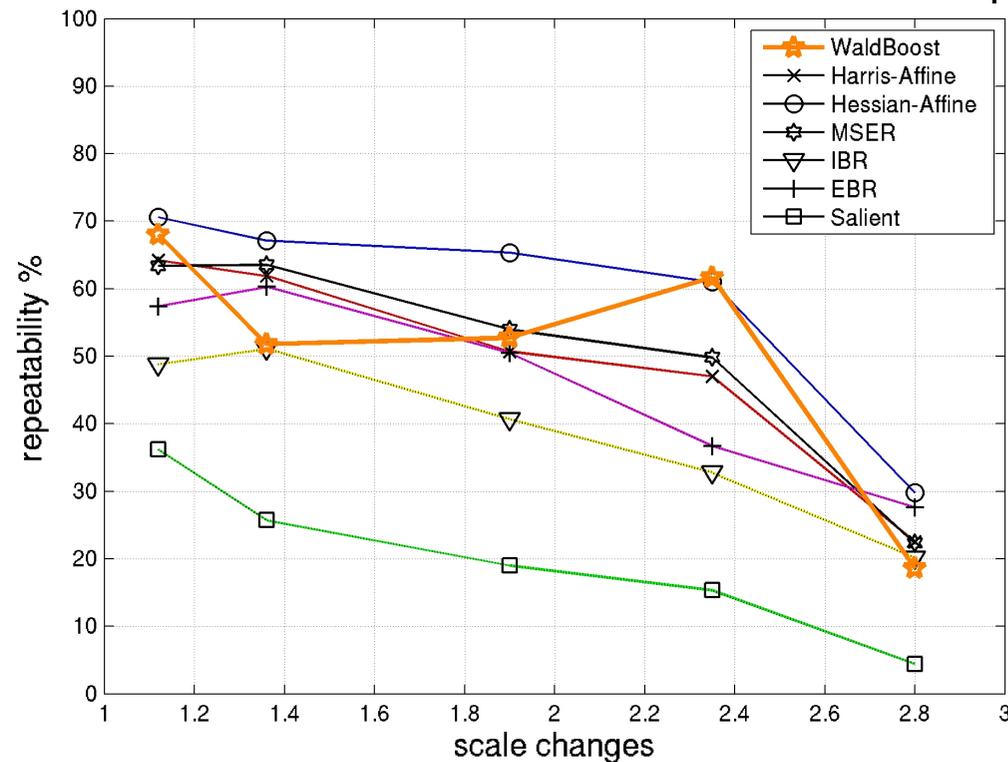


- Yellow: repeated detections - 85%
- Red: original detections not found by the WaldBoost detector

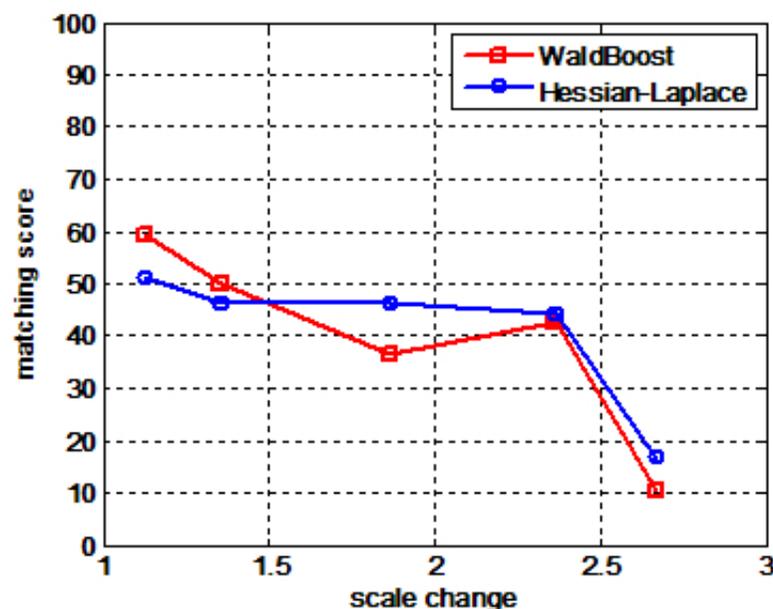
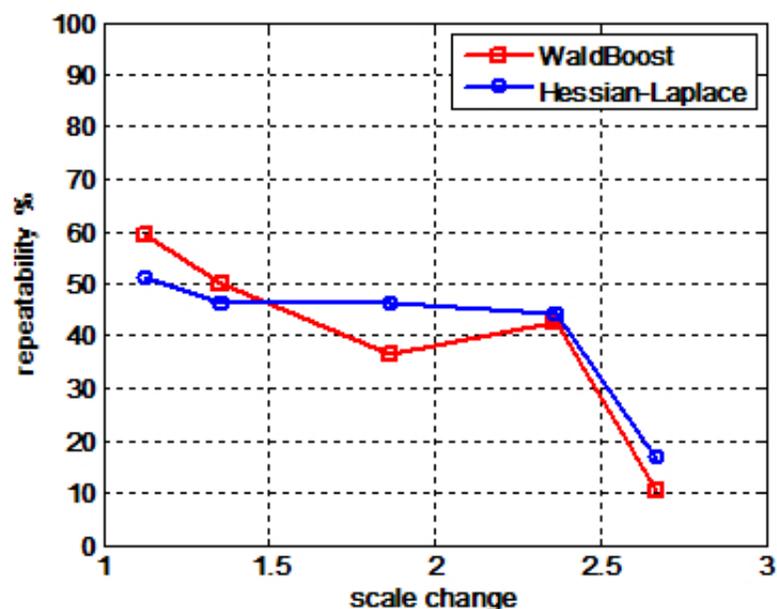
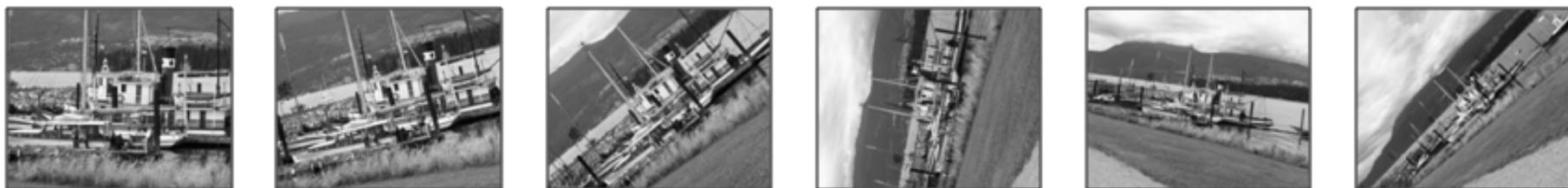
Emulating Harris-Laplace Detector - Results



Repeatability score of the WaldBoost detector for Bark sequence (overlap 40%, norm. size = 30 pixels).

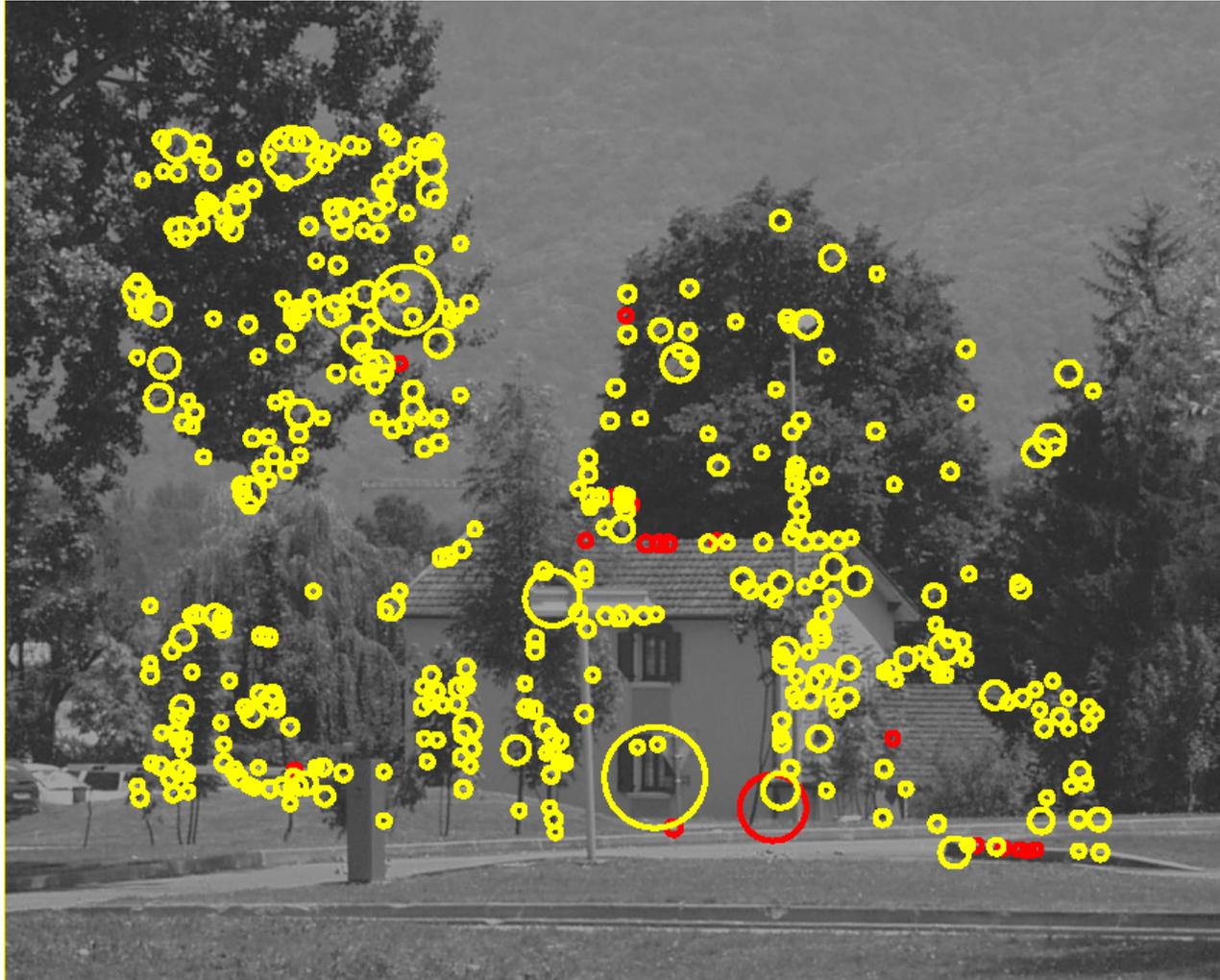


Repeatability score of the WaldBoost detector for Boat sequence (overlap 40%, norm. size = 30 pixels).

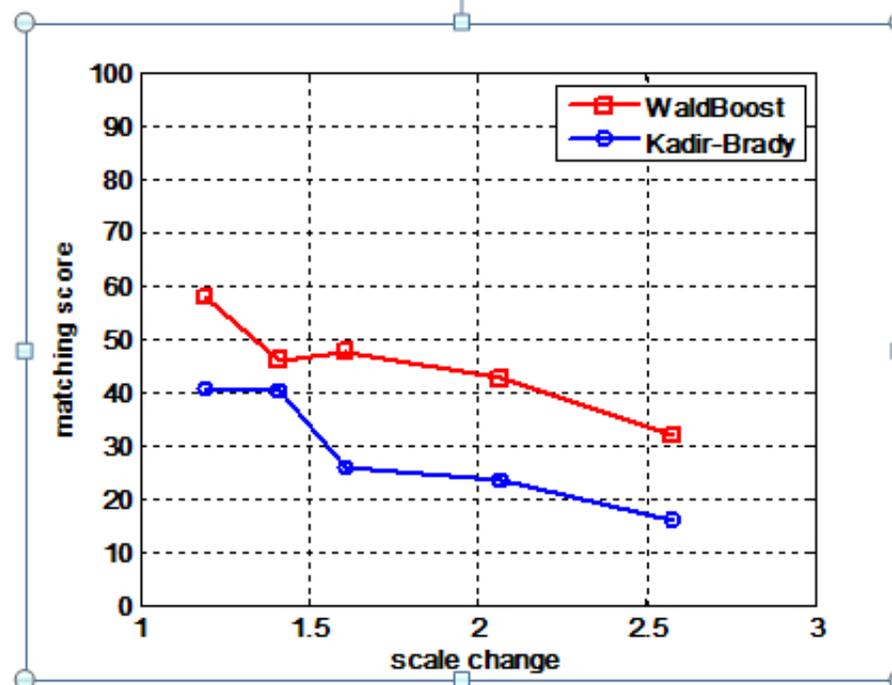
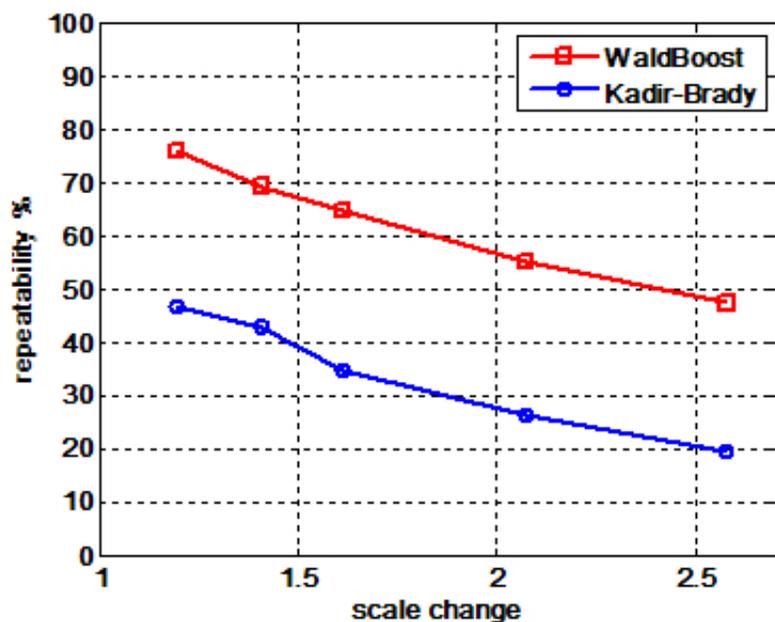
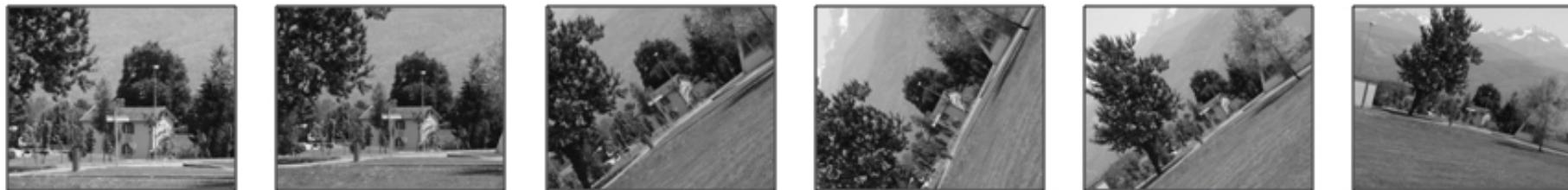


Waldboost performance:

- Higher number of correspondences (not shown).
- Higher or similar percentage of correct matches.
- Similar speed. For a 850x680 image:
Original: 1.3 secs v. WaldBoost: 0.7 secs



- Yellow: repeated dections - 96%
- Red: original detections not found by the WaldBoost detector



Waldboost performance:

- Higher number of correspondences and correct matches.
- Significant speed-up ($\approx 10^2$). For a 850x680 image:
Original: 1 min 44 sec v. WaldBoost: 0.76 sec

Applying Wald #3: Sequential Correspondence Selection by Cosegmentation (Cech, Matas, Perdoch, PAMI 2009)



36/46

Establishing correspondences, Standard solution (Lowe, IJCV 2004):
(steps 1. and 2 carried out in both images independently)

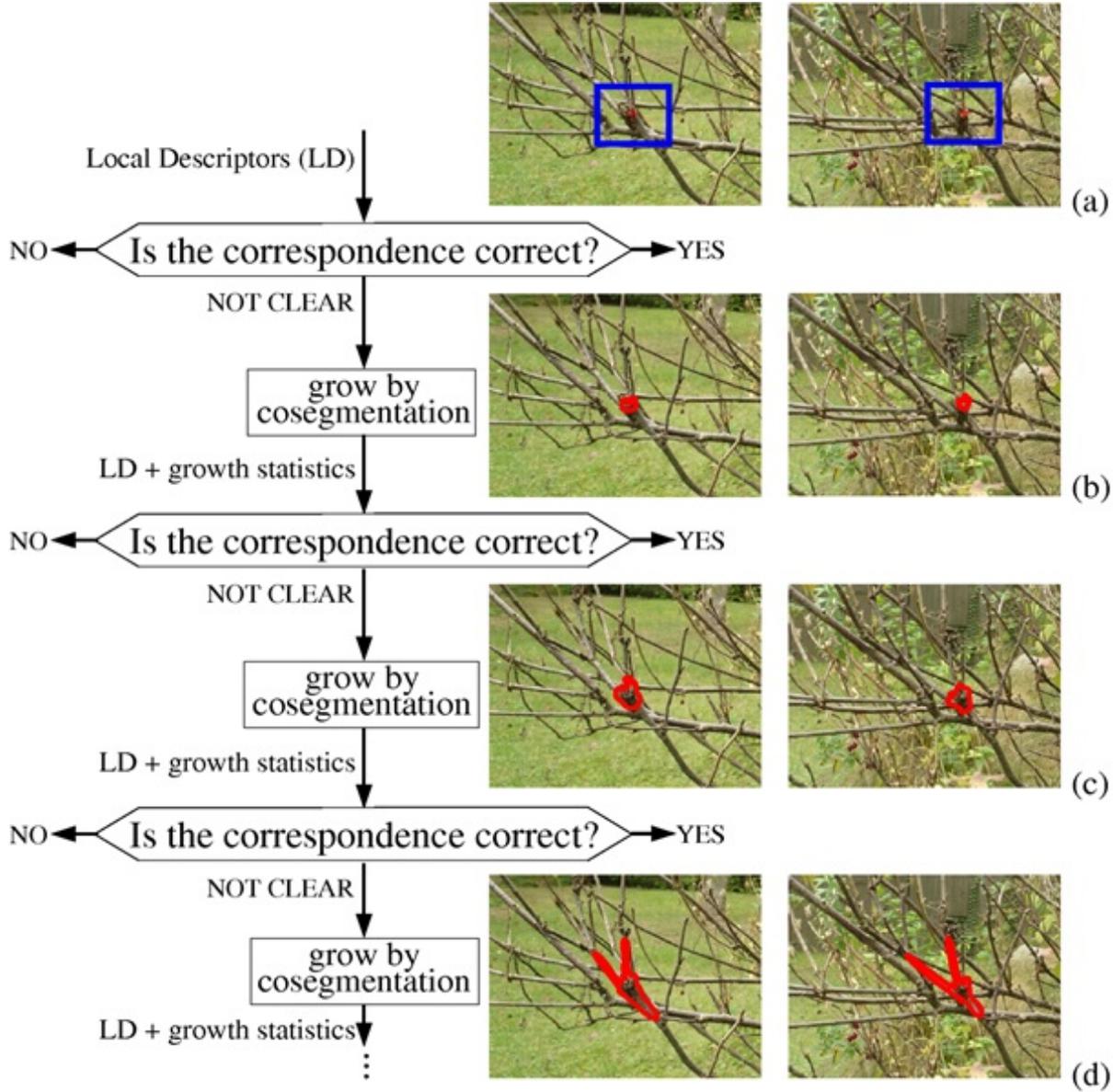
1. Detect covariant regions (Diff. of Gaussians, Hessian, MSER, Harris, ...)
2. Describe regions with a SIFT from an "arbitrary measurement area"
3. Select regions matches with SIFT ratio ≤ 0.8



Drawbacks:

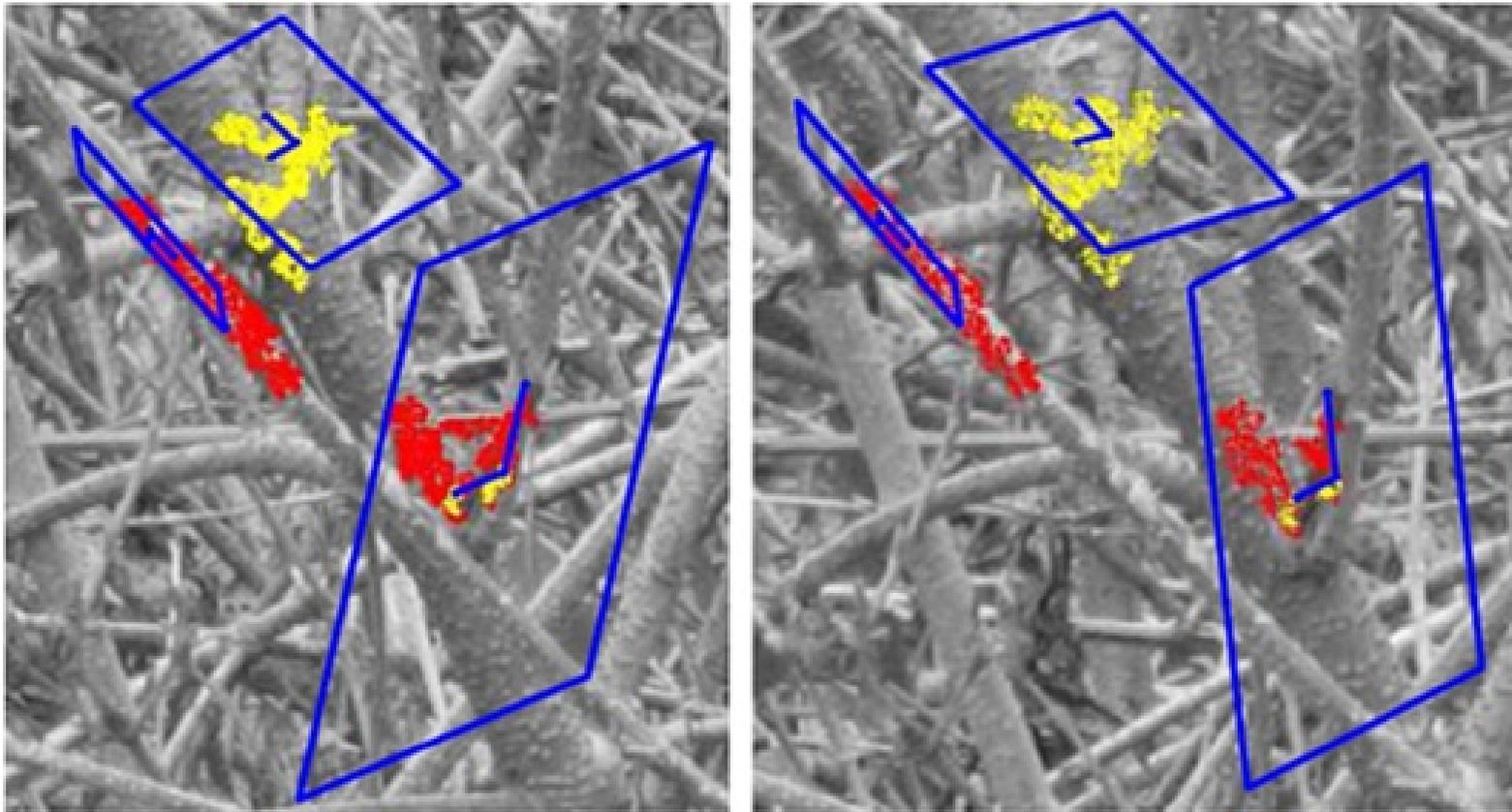
1. fix-shaped measurement regions
2. independent detection, normalization (geometric, photometric)

Sequential Correspondence Verification (SCV) by Cosegmentation



The method (similar to stereo matching or cosegmentation):

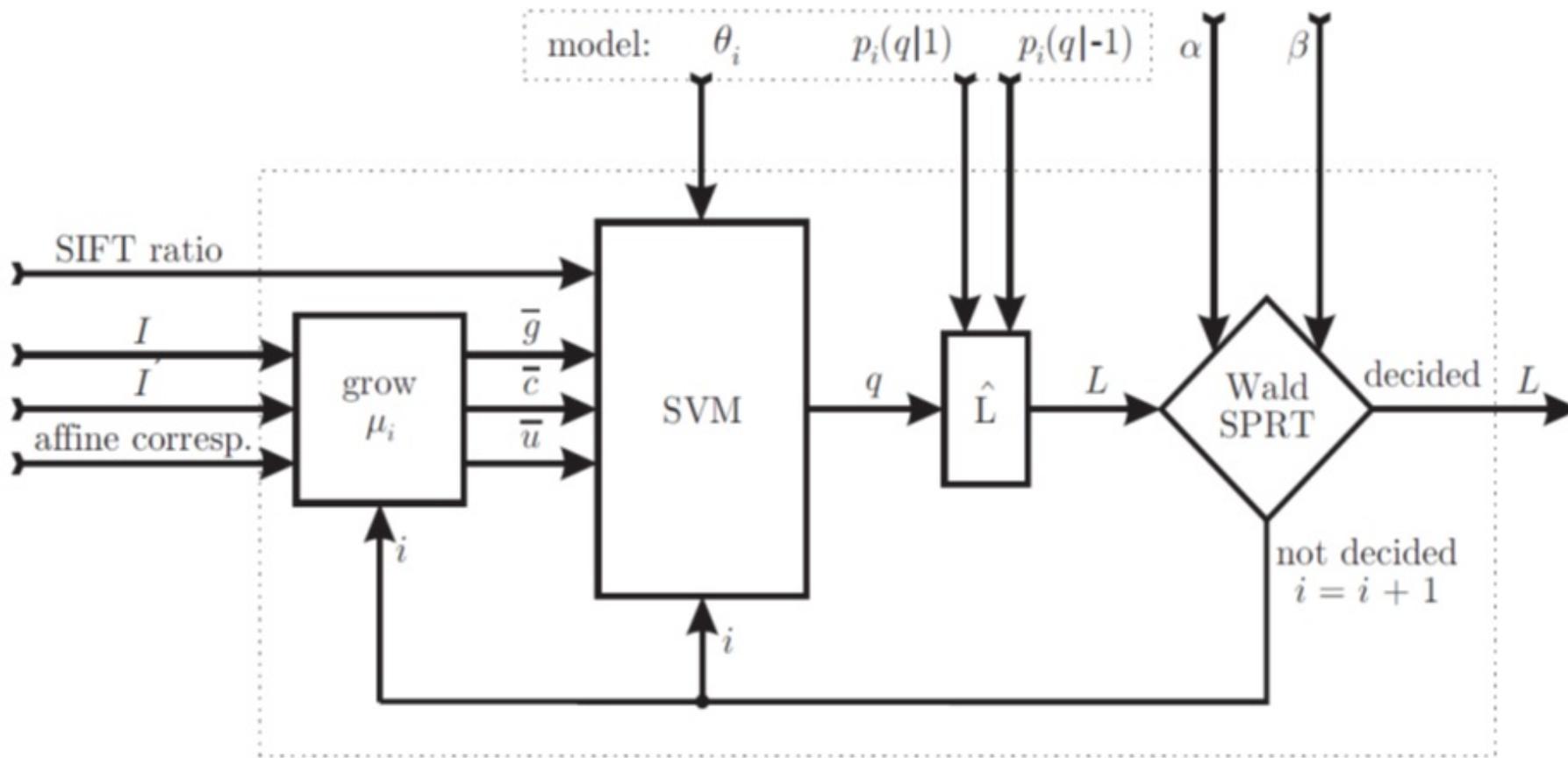
1. Initial matching seeds from SIFT correspondence
2. Apply a match growing process with a fixed number of steps
3. Collect statistics from the growth process:
growth rate, average correlation, non-bijection
4. Use Wald's SPRT test to accept or reject the correspondence



Sequential Correspondence Verification (SCV)



- Growing the correspondence collects more data, increases the discriminability.
- It costs time: a suitable task for sequential decision.

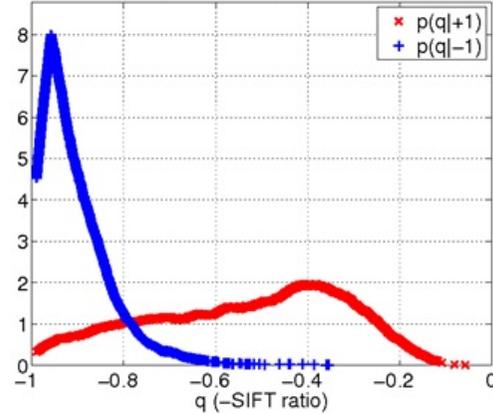


SCV - training

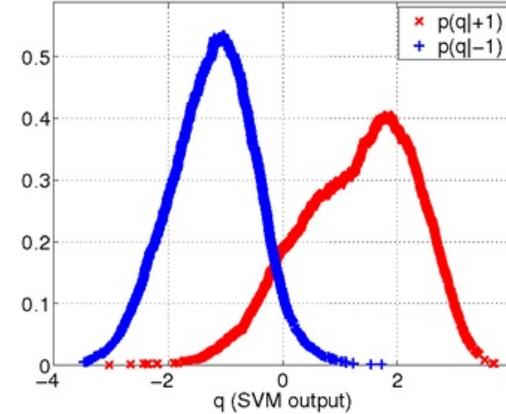


Distribution of the statistics for correct and incorrect correspondences learned from a training set of .

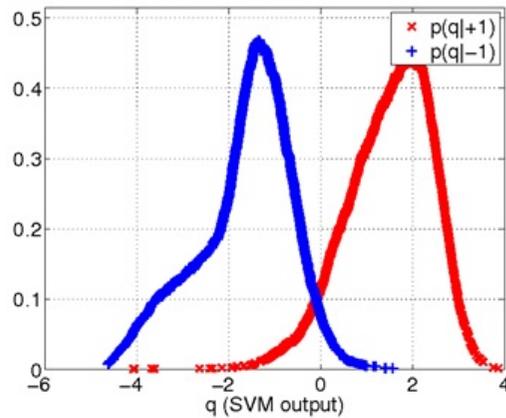
$i = 1 : \mu = 0$ (SIFT ratio only)



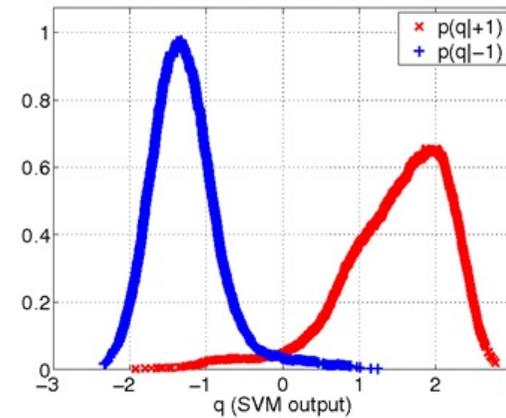
$i = 2 : \mu = 10$



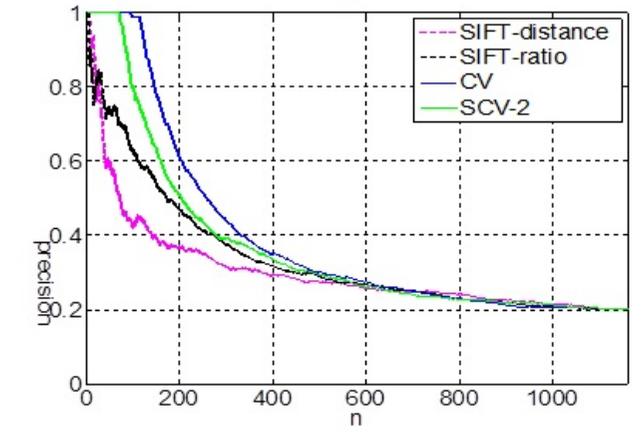
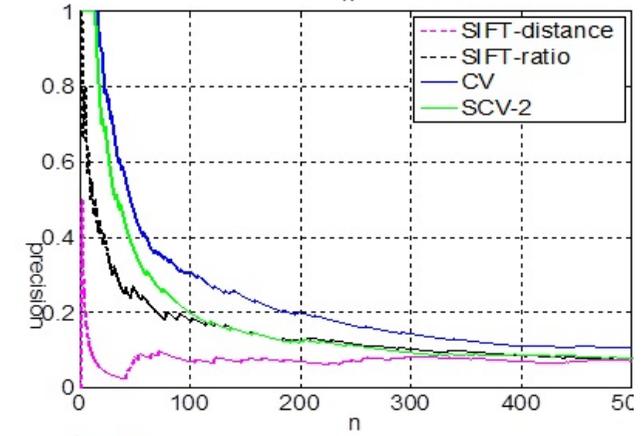
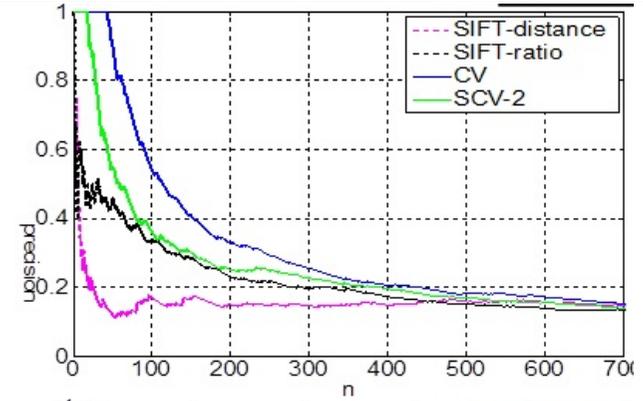
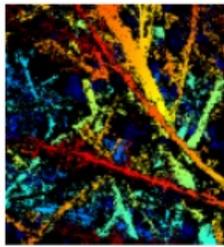
$i = 3 : \mu = 100$



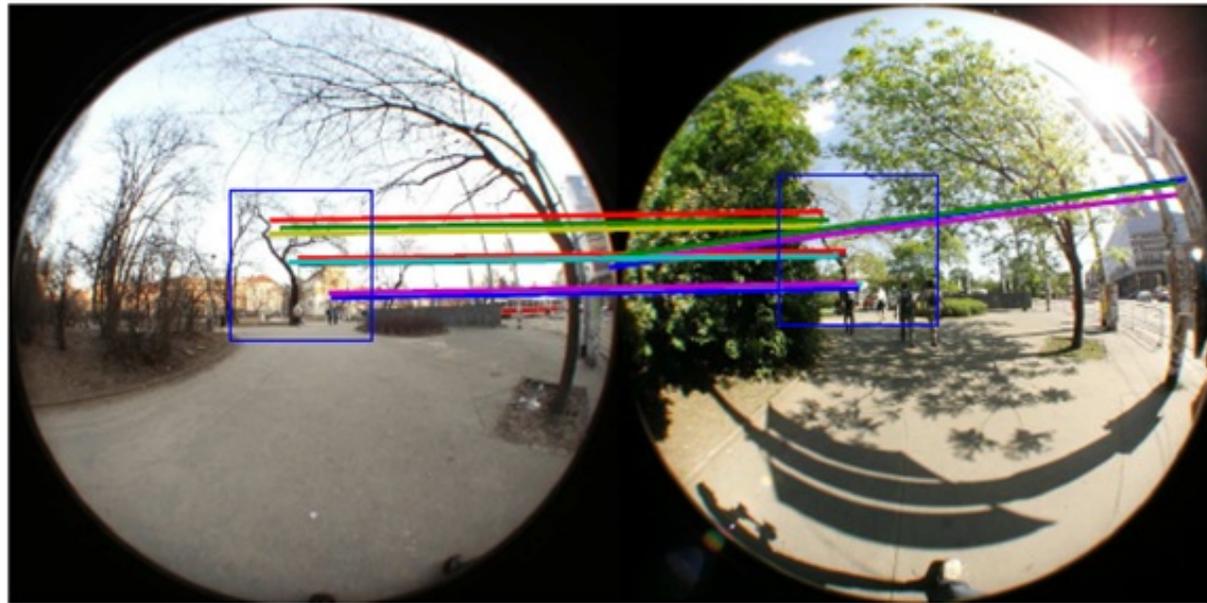
$i = 4 : \mu = 1000$



SCV - Results 1



SCV - Results 2



zoom (grown regions)



- high precision and recall, significantly outperforms standard SIFT-ratio test
- very fast (0.5 sec / 1000 correspondences), controllable by accuracy requirements
- SCV time is small in comparison with the time spent by matching of tentative correspondences
- applicable before RANSAC
- the process generating tentative correspondences can be much more permissive: even 99% of outliers not a problem, correct correspondences recovered

Implementation available for download at <http://cmp.felk.cvut.cz/software/SCV>

Repeat $k/(1-\alpha)$ times

1. Hypothesis generation
2. Model pre-verification $T_{d,d}$ test
 - Verify $d \ll N$ data points, reject the model if not all d data points are consistent with the model
3. Model verification
Verify the rest of the data points

Time
 t_M

$\bar{m}_s \cdot V$

V – average number of data points verified

α – probability that a good model is rejected by $T_{d,d}$ test

$$t = \frac{k}{1 - \alpha} (t_M + \bar{m}_s V)$$

Repeat $k/(1-1/A)$ times

1. Hypothesis generation

2. Model verification

use SPRT

Time

t_M

$\bar{m}_S \cdot C \log A$

$$C \approx \left((1 - \delta) \log \frac{1 - \delta}{1 - \varepsilon} + \delta \log \frac{\delta}{\varepsilon} \right)^{-1}$$

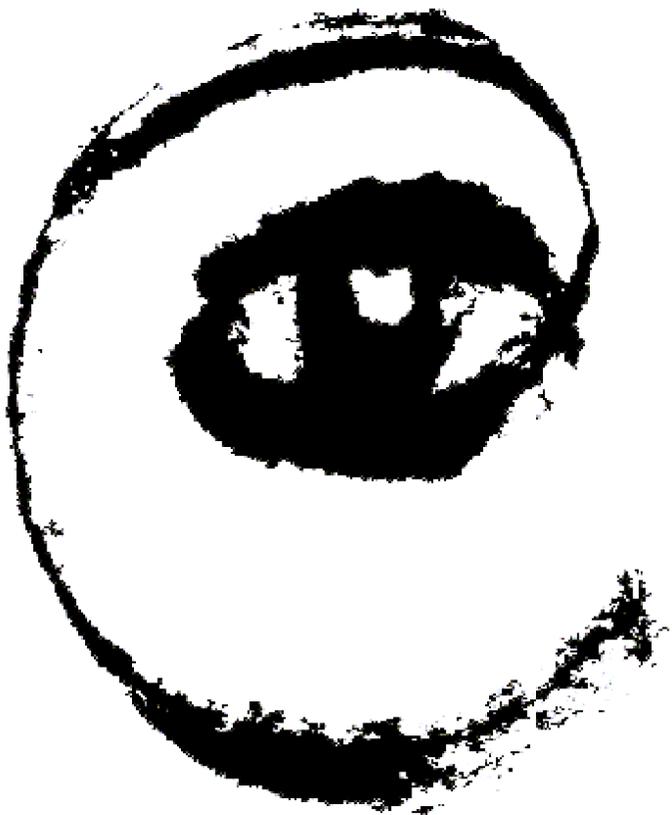
$$t(A) = \frac{k}{(1 - 1/A)} (t_M + \bar{m}_S C \log A)$$

In sequential statistical decision problem decision errors are traded off for time. These are two incomparable quantities, hence the constrained optimization.

In WaldSAC, decision errors cost time (more samples) and there is a single minimised quantity, time $t(A)$, a function of a single parameter A .



- Time to decision vs. precision trade-off problem was formalised as a constrained optimisation problem.
- Wald's SPRT can be used directly in some computer vision problems
- WaldBoost overcomes limitations of SPRT to a priori ordered measurements and known multidimensional pdf's by using the AdaBoost algorithm.
- Adaboost learning converges (on an infinite training set) to the optimal decision function, but the ordering of weak classifiers is greedy.
- WaldBoost has been successfully applied in a number of applications
 - faces, vehicles, licence plates, ,
 - tracking (Grabner et al. ICPR 2008)
- The order of weak classifiers is fixed, selection of h_t depending on the *values* x_1, x_2, \dots, x_{k-1} not (yet) considered (leads to a decision tree with Wald's test)



m p

Segmentation as Selective Search for Object Recognition

Koen E. A. van de Sande* Jasper R. R. Uijlings† Theo Gevers* Arnold W. M. Smeulders*
*University of Amsterdam †University of Trento
Amsterdam, The Netherlands Trento, Italy

ksande@uva.nl, jrr@disi.unitn.it, th.gevers@uva.nl, a.w.m.smeulders@uva.nl

Abstract

For object recognition, the current state-of-the-art is based on exhaustive search. However, to enable the use of more expensive features and classifiers and thereby progress beyond the state-of-the-art, a selective search strategy is needed. Therefore, we adapt segmentation as a selective search by reconsidering segmentation: We propose to generate many approximate locations over few and precise object delineations because (1) an object whose location is never generated can not be recognised and (2) appearance and immediate nearby context are most effective for object recognition. Our method is class-independent and is shown to cover 96.7% of all objects in the Pascal VOC 2007 test set using only 1,536 locations per image. Our selective search enables the use of the more expensive bag-of-words method which we use to substantially improve the state-of-the-art by up to 8.5% for 8 out of 20 classes on the Pascal VOC 2010 detection challenge.

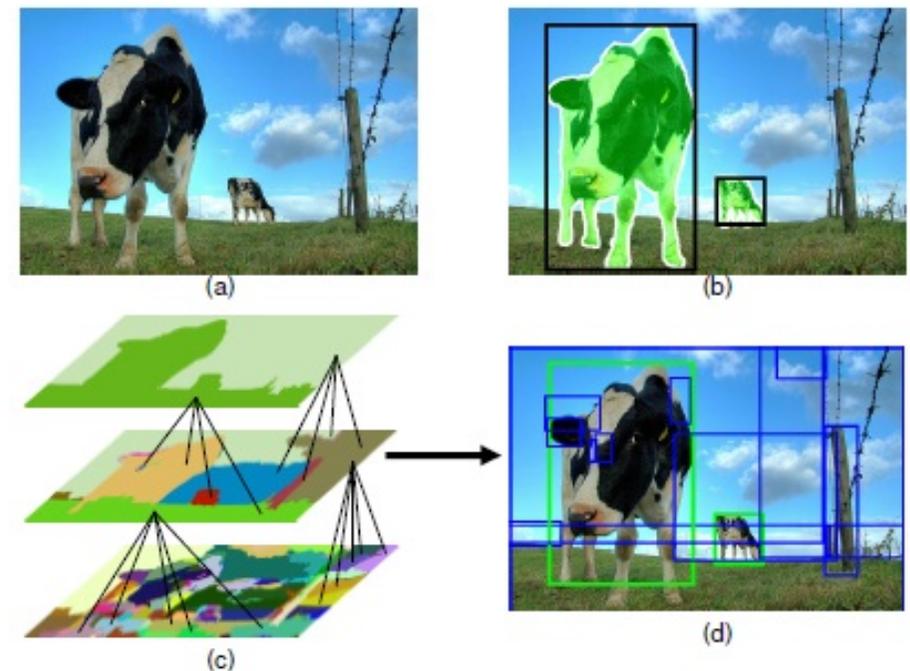
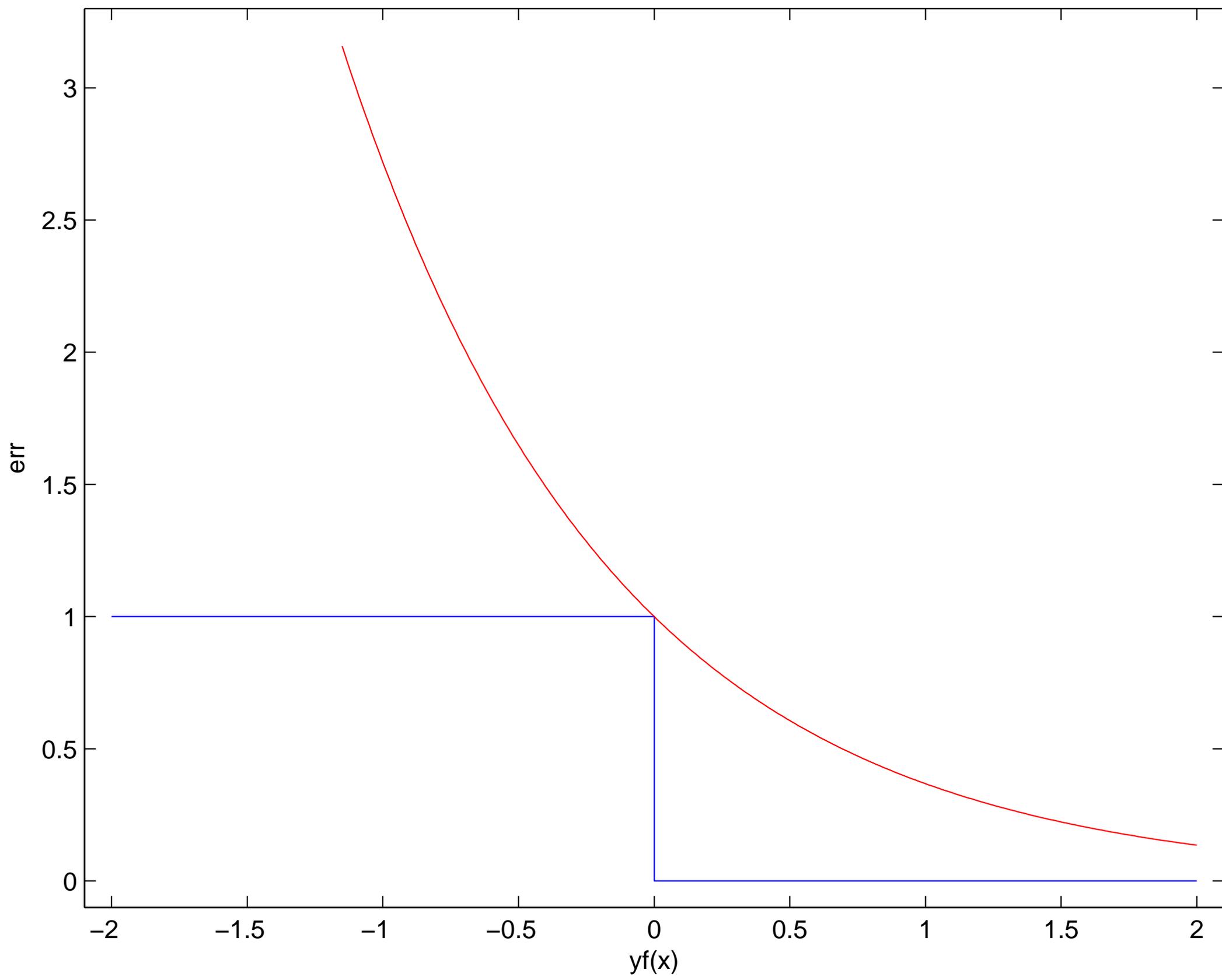
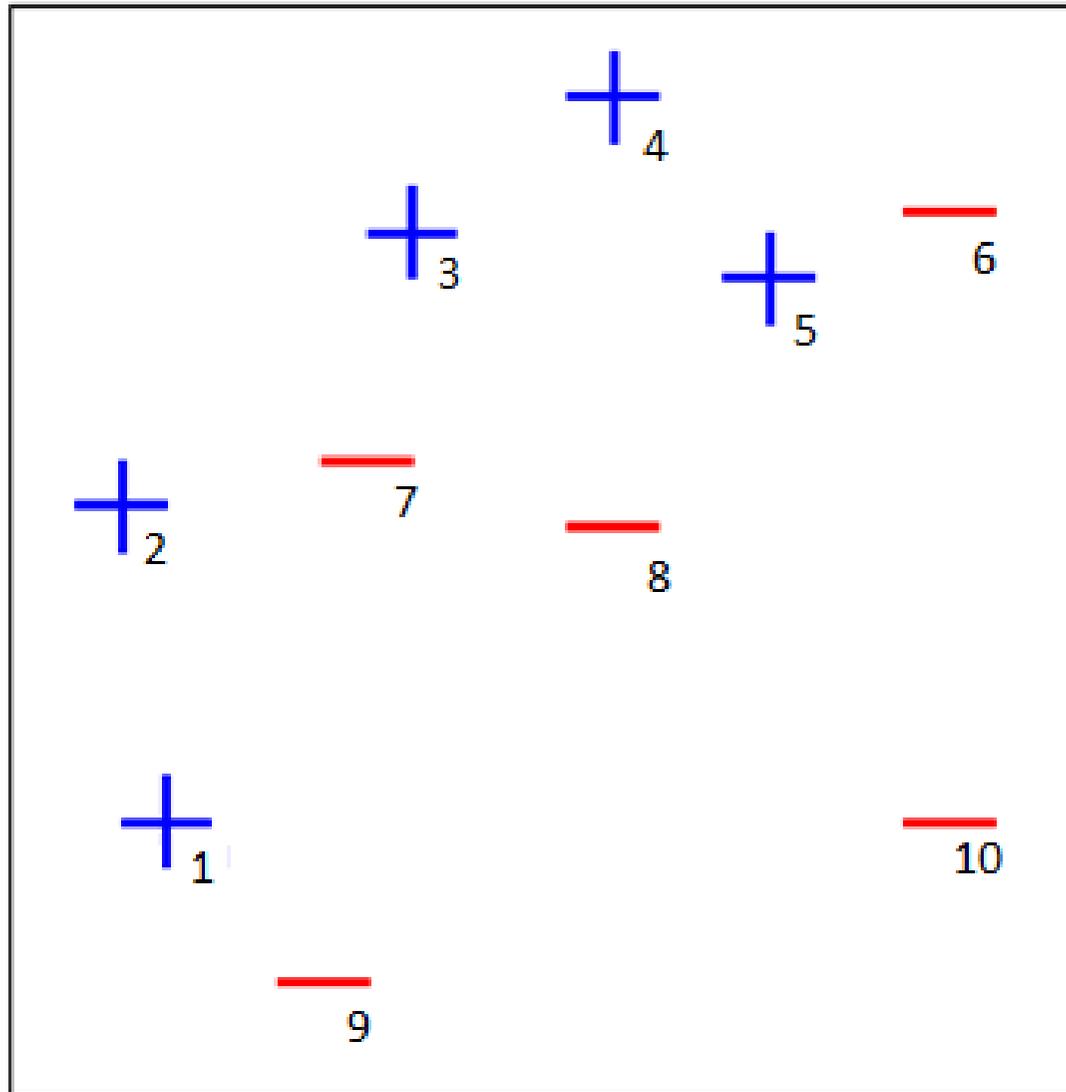
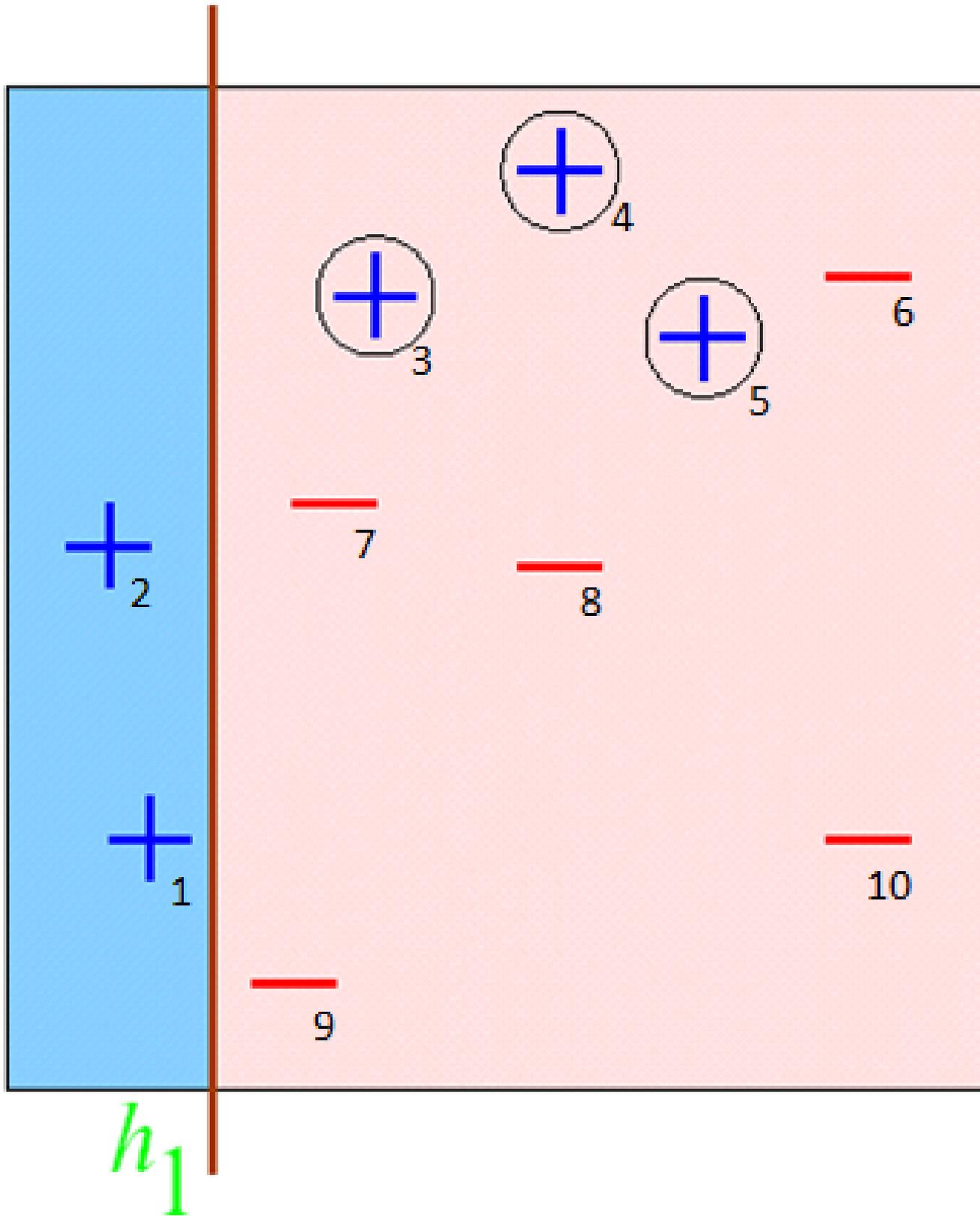


Figure 1. Given an image (a) our aim is to find its objects for which the ground truth is shown in (b). To achieve this, we adapt segmentation as a selective search strategy: We aim for high recall by generating locations at all scales and account for many different scene conditions by employing multiple invariant colour spaces. Example object hypotheses are visualised in (d).



D_1





$$\varepsilon_1 = 0.30$$

$$\alpha_1 = 0.42$$

D_2

$+$
2

$-$
7

$-$
8

$-$
6

$+$
4

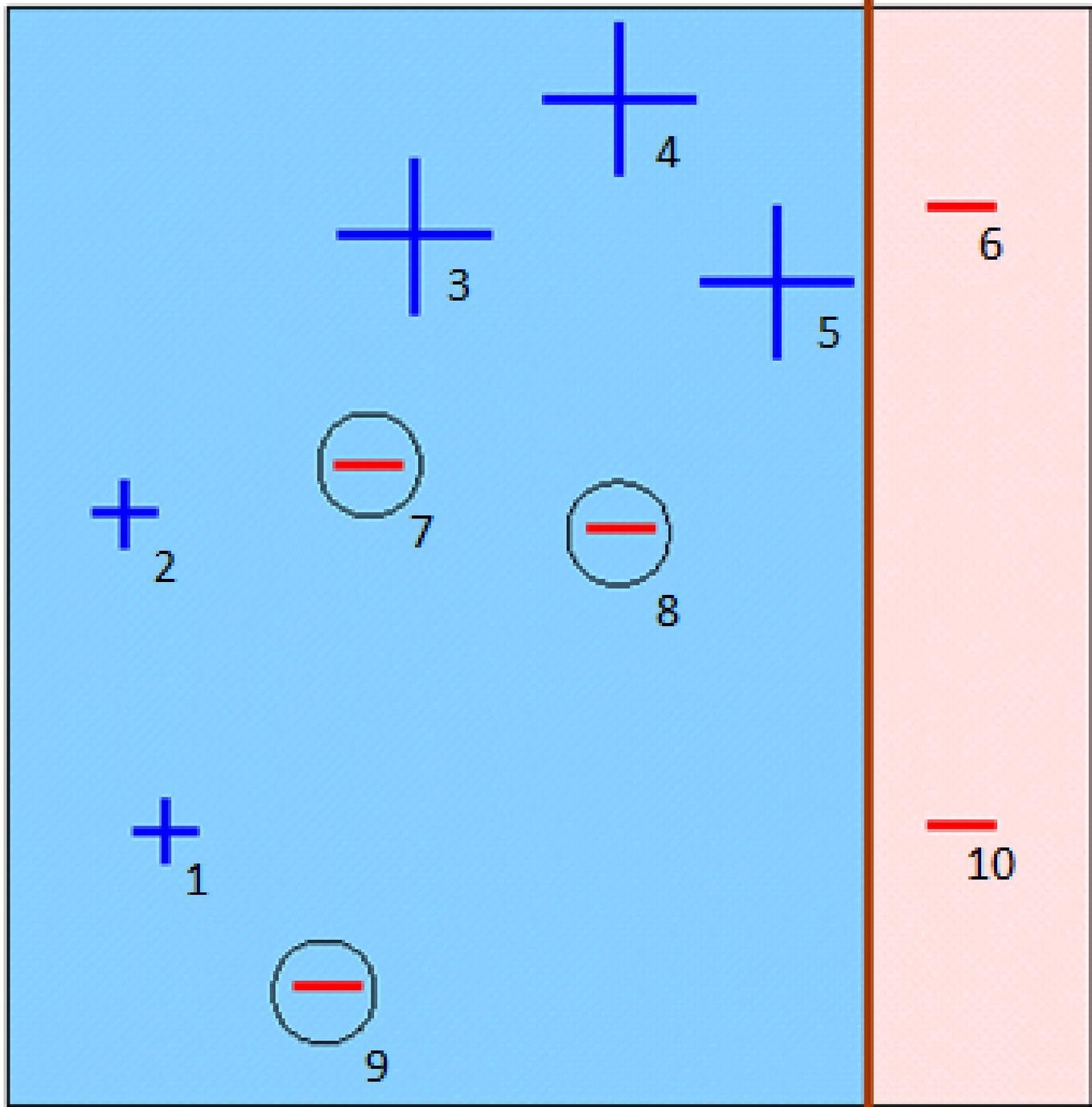
$+$
3

$+$
5

$+$
1

$-$
9

$-$
10

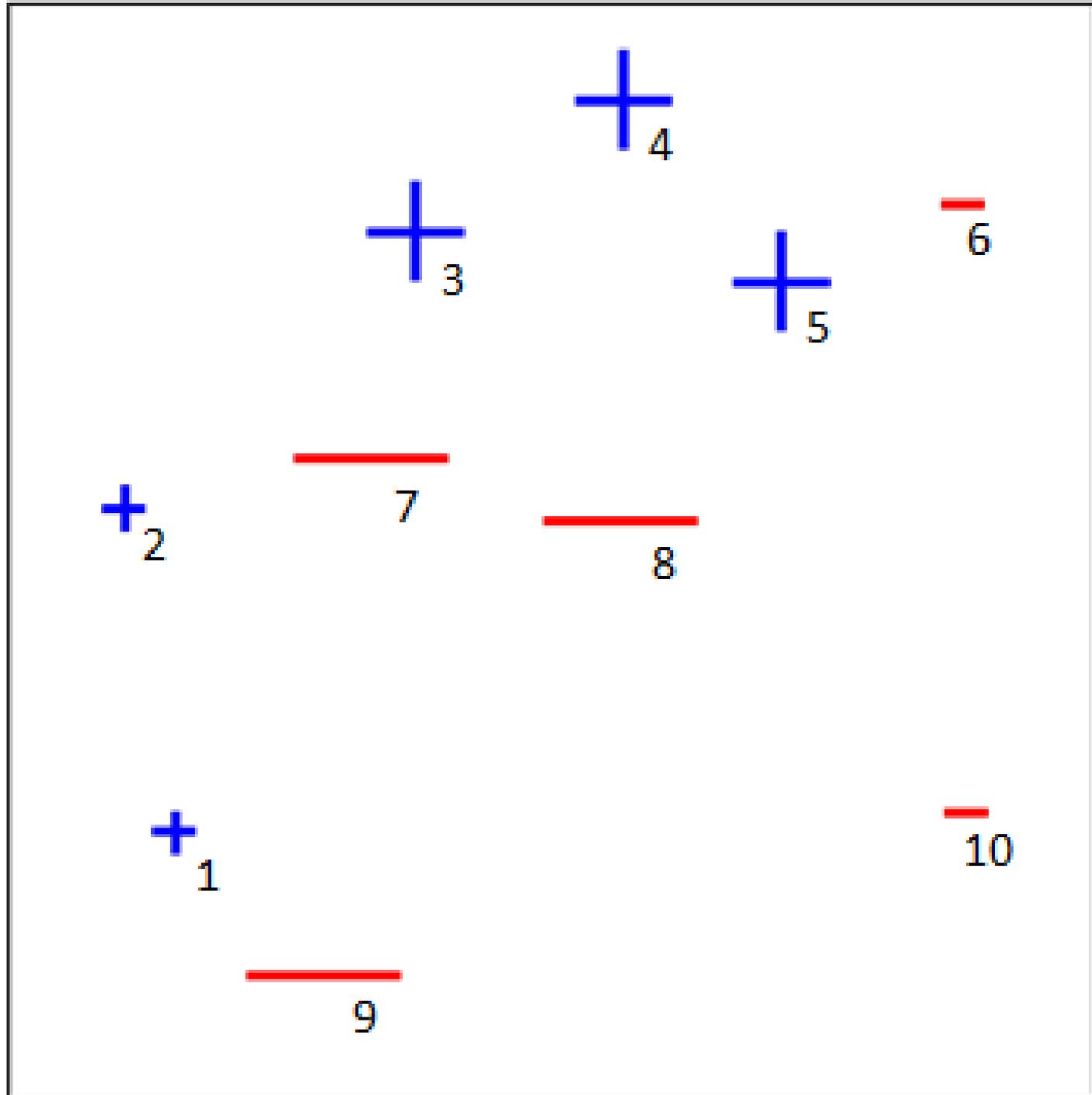


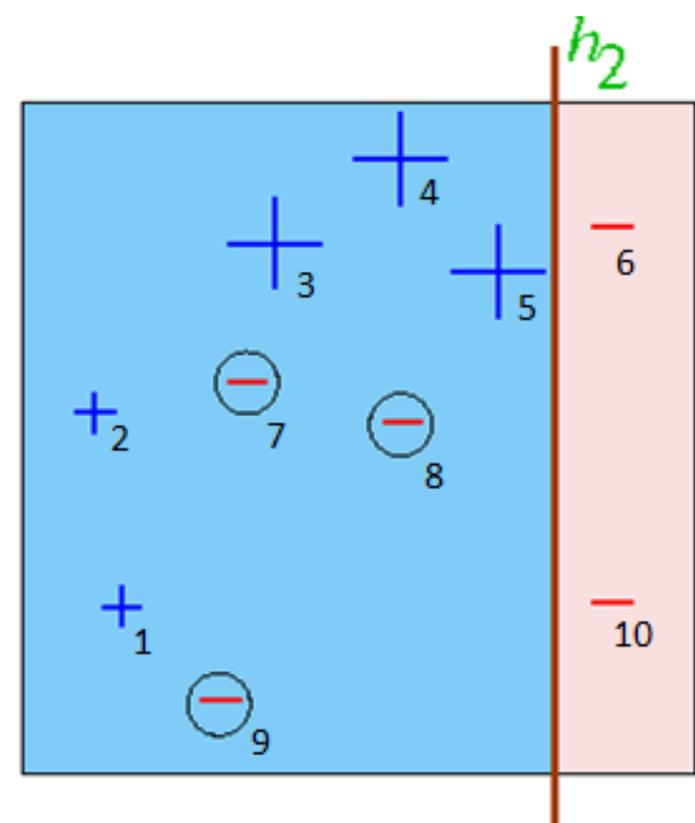
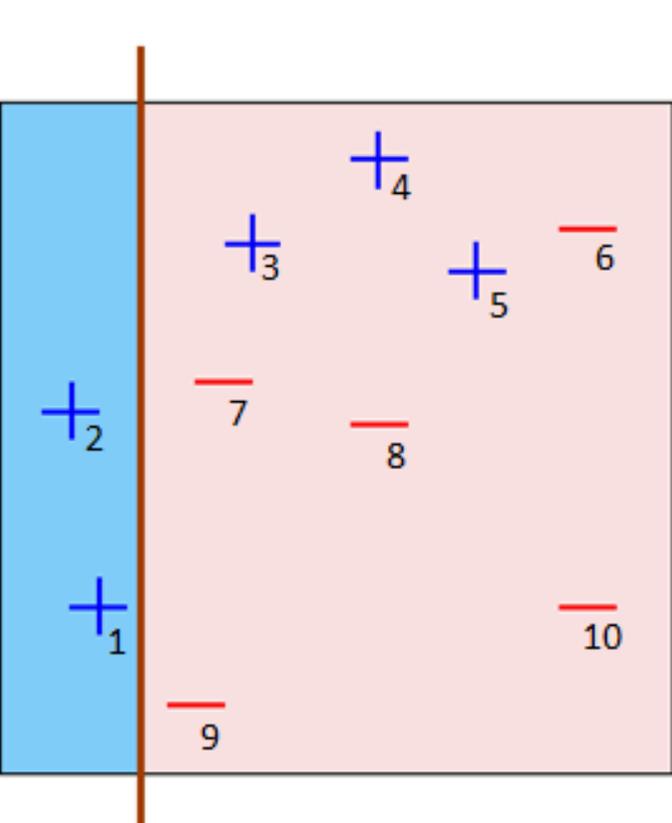
$$\epsilon_2 = 0.21$$

$$\alpha_2 = 0.65$$

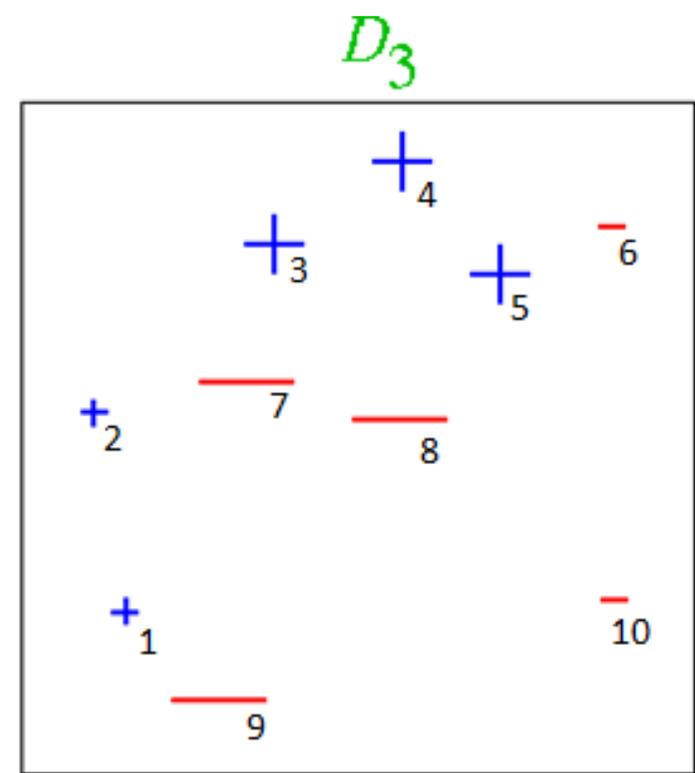
h_2

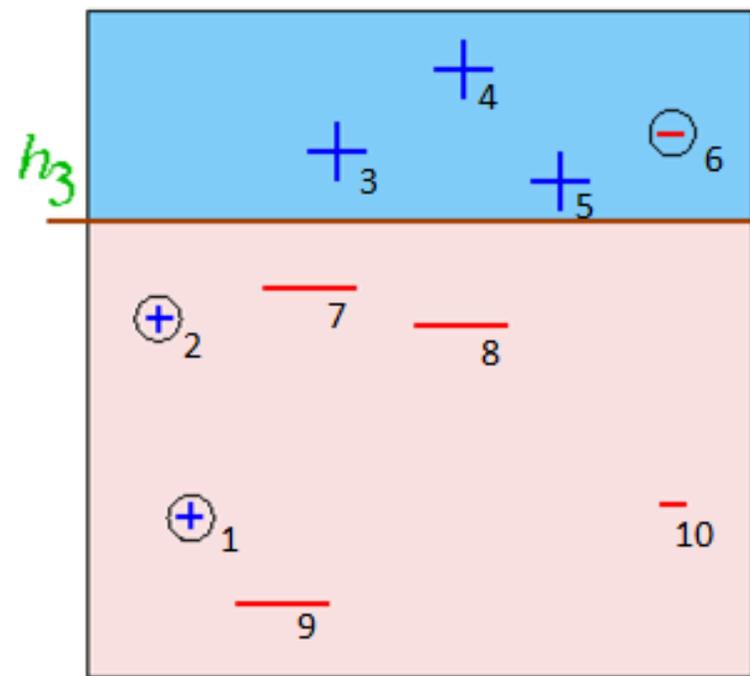
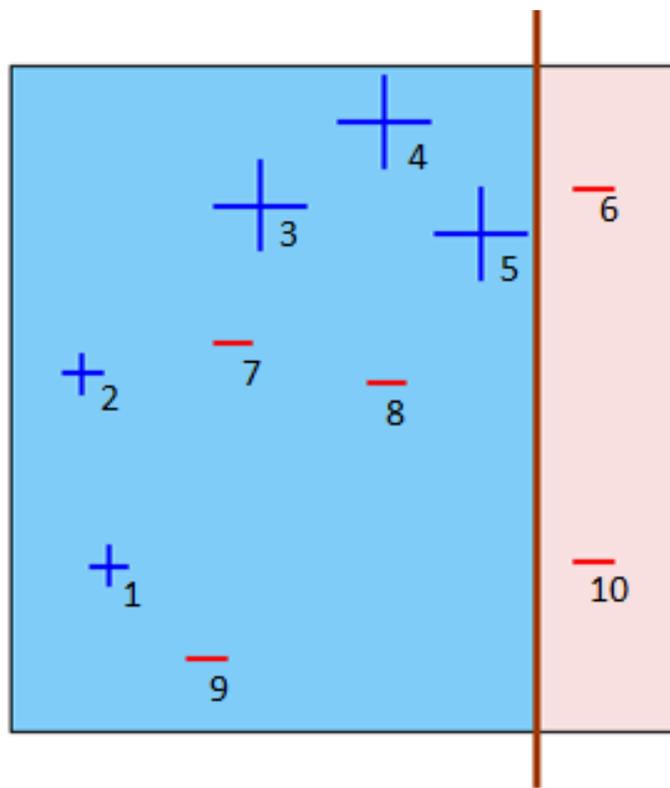
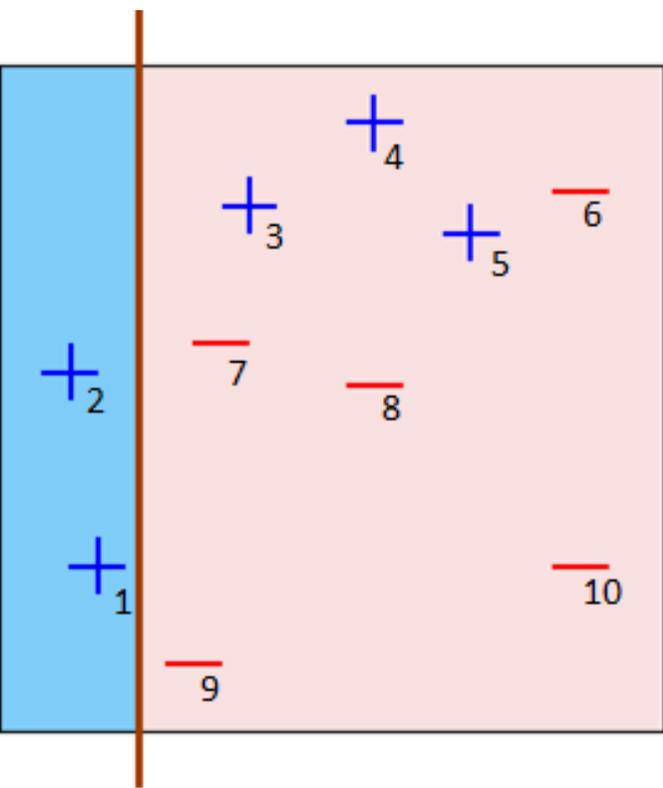
D_3





$\epsilon_2 = 0.21$
 $\alpha_2 = 0.65$

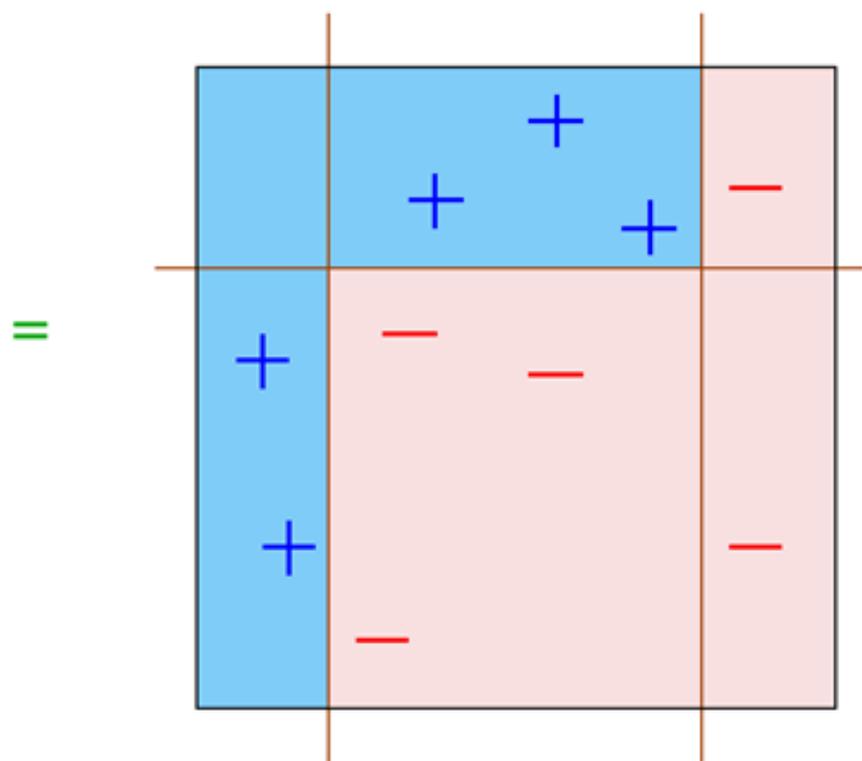


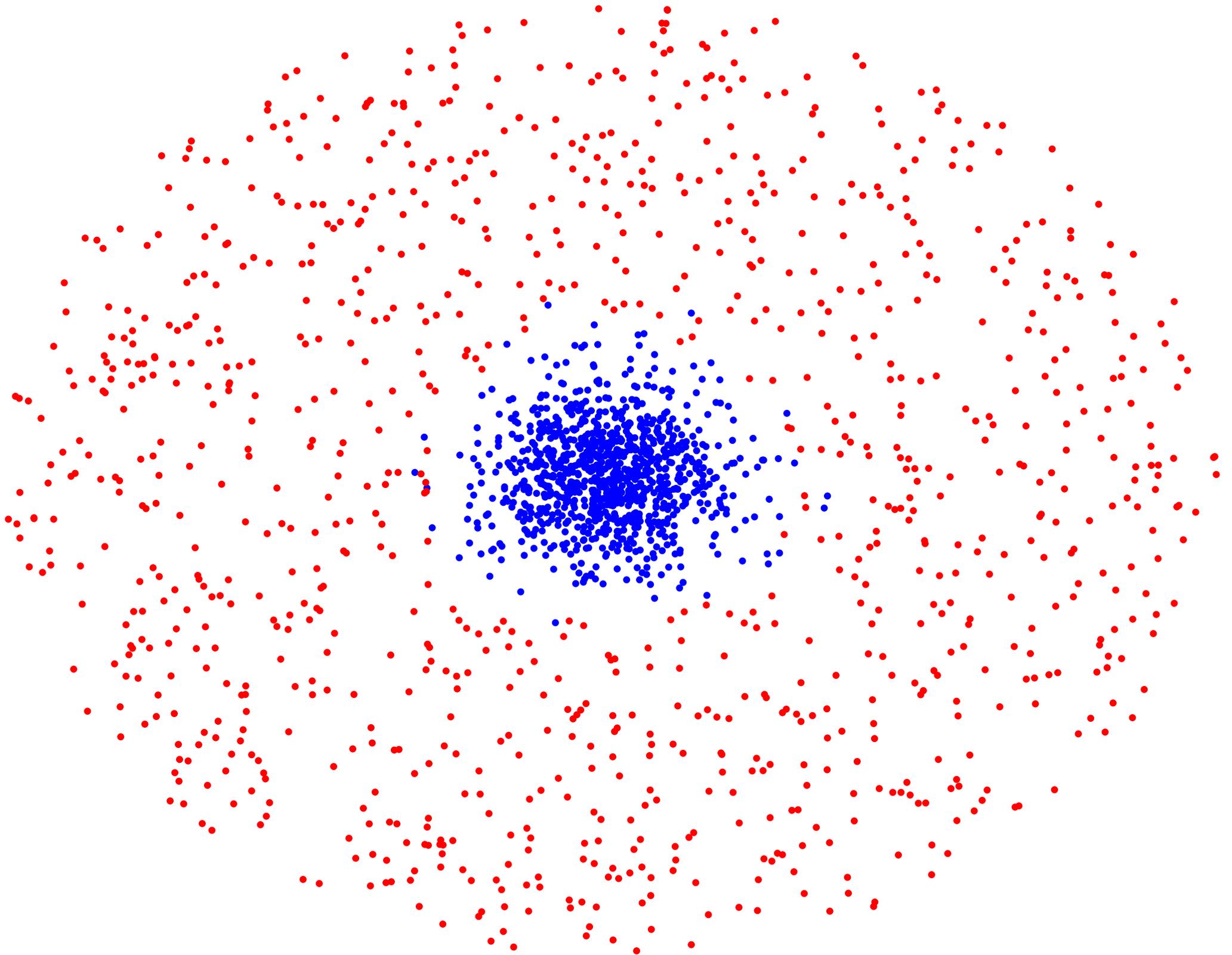


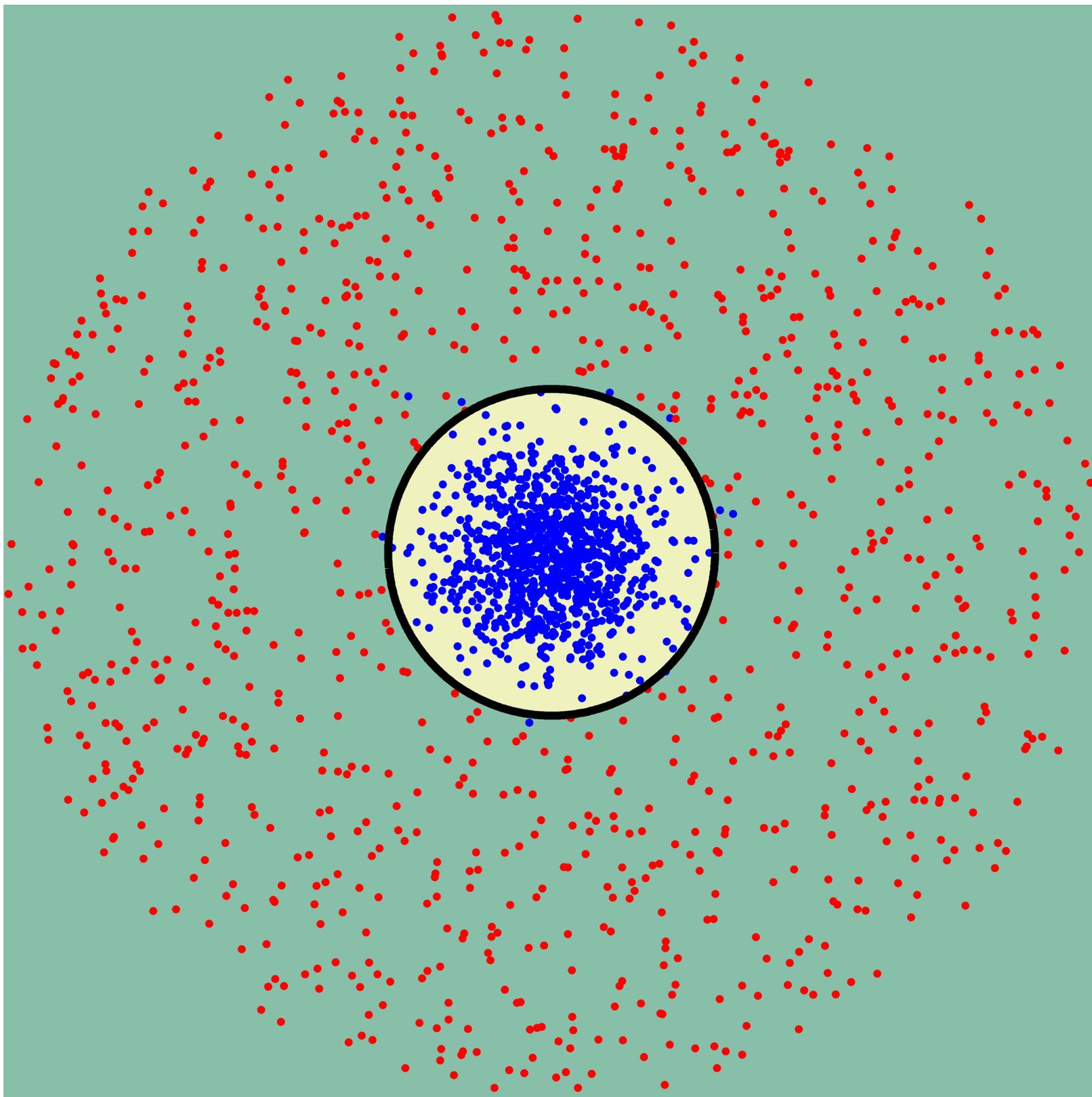
$$\epsilon_3 = 0.14$$

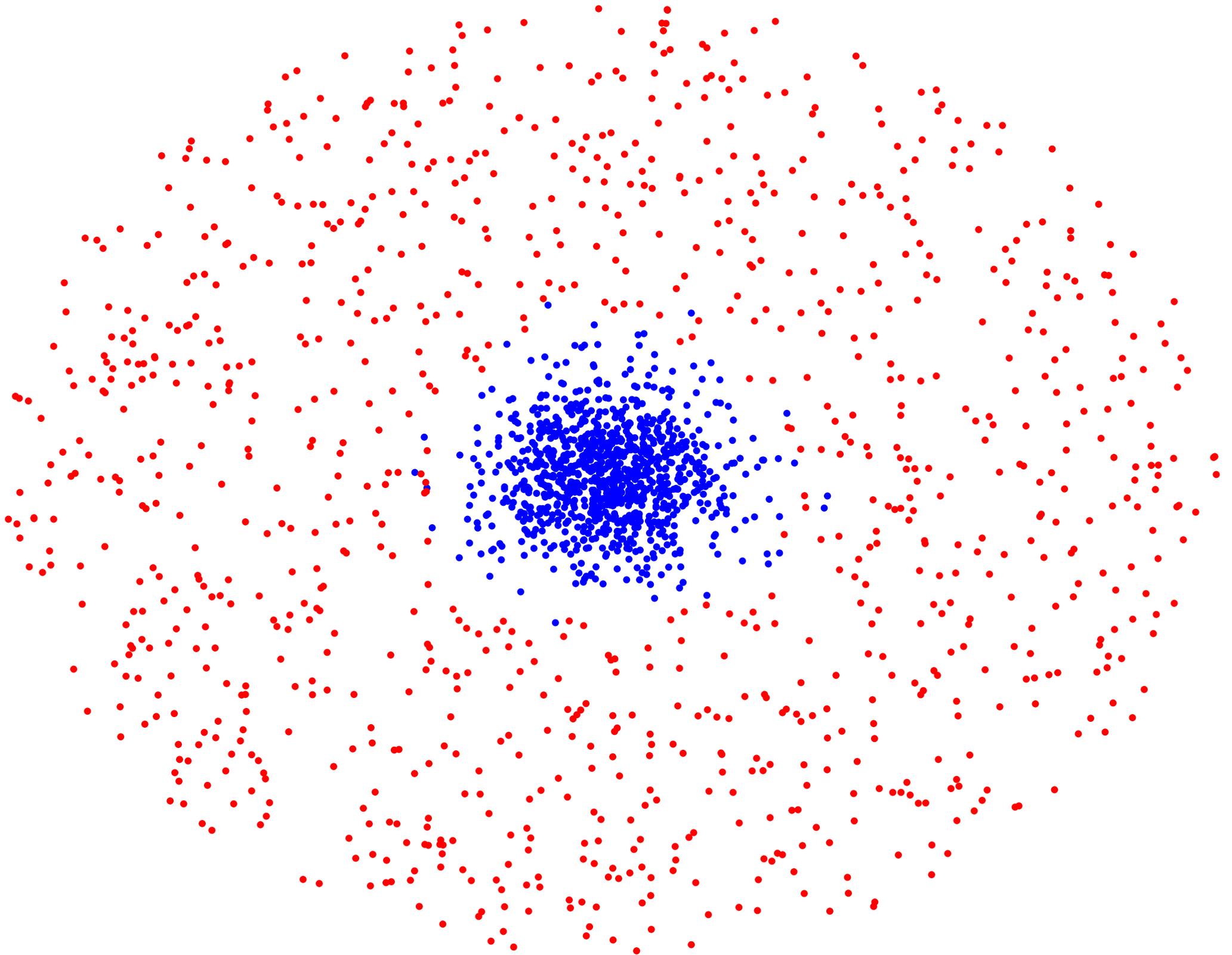
$$\alpha_3 = 0.92$$

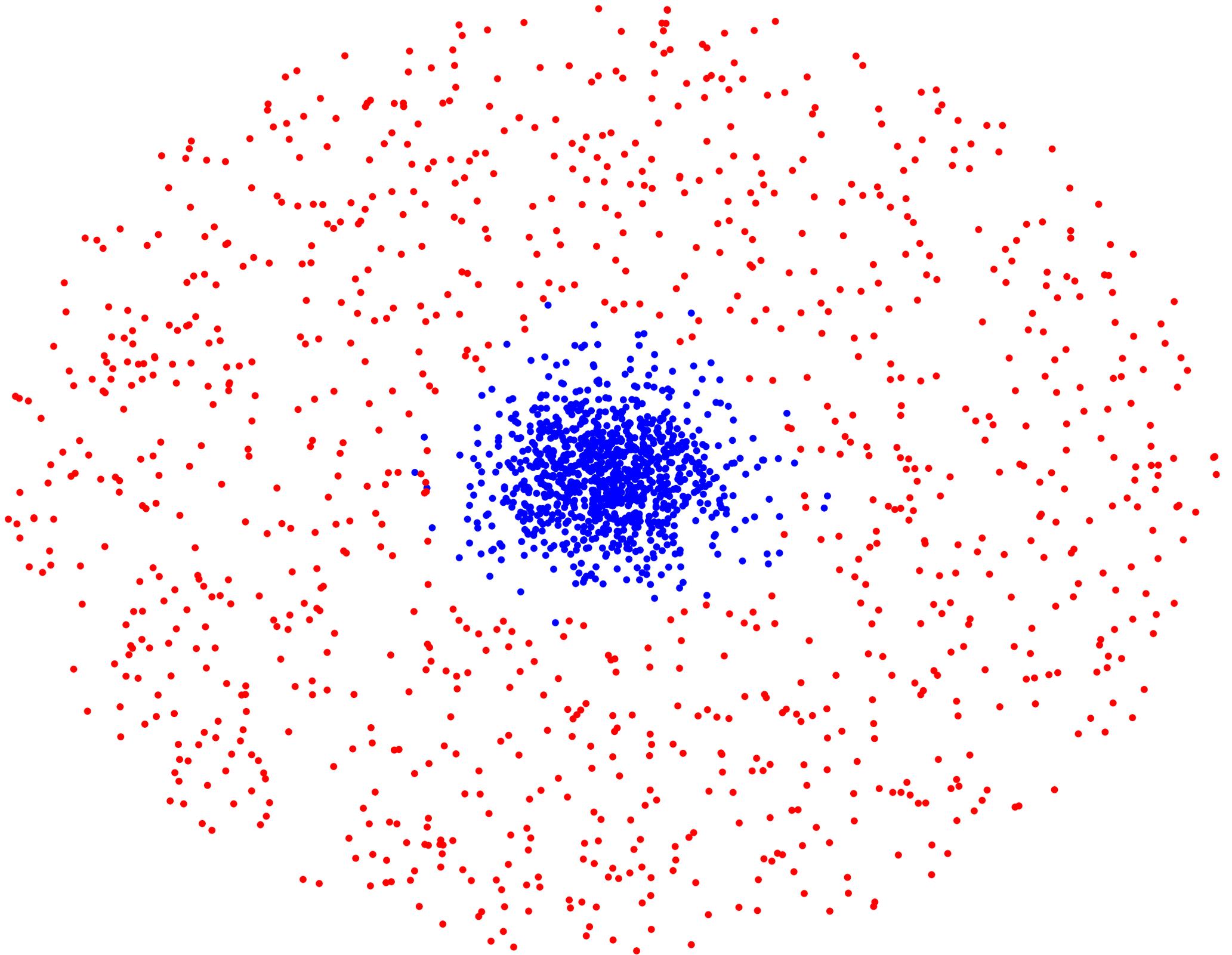
$$H_{\text{final}} = \text{sign} \left(0.42 \begin{array}{|c|} \hline \text{blue} \\ \hline \end{array} + 0.65 \begin{array}{|c|} \hline \text{blue} \\ \hline \end{array} + 0.92 \begin{array}{|c|} \hline \text{blue} \\ \hline \end{array} \right)$$

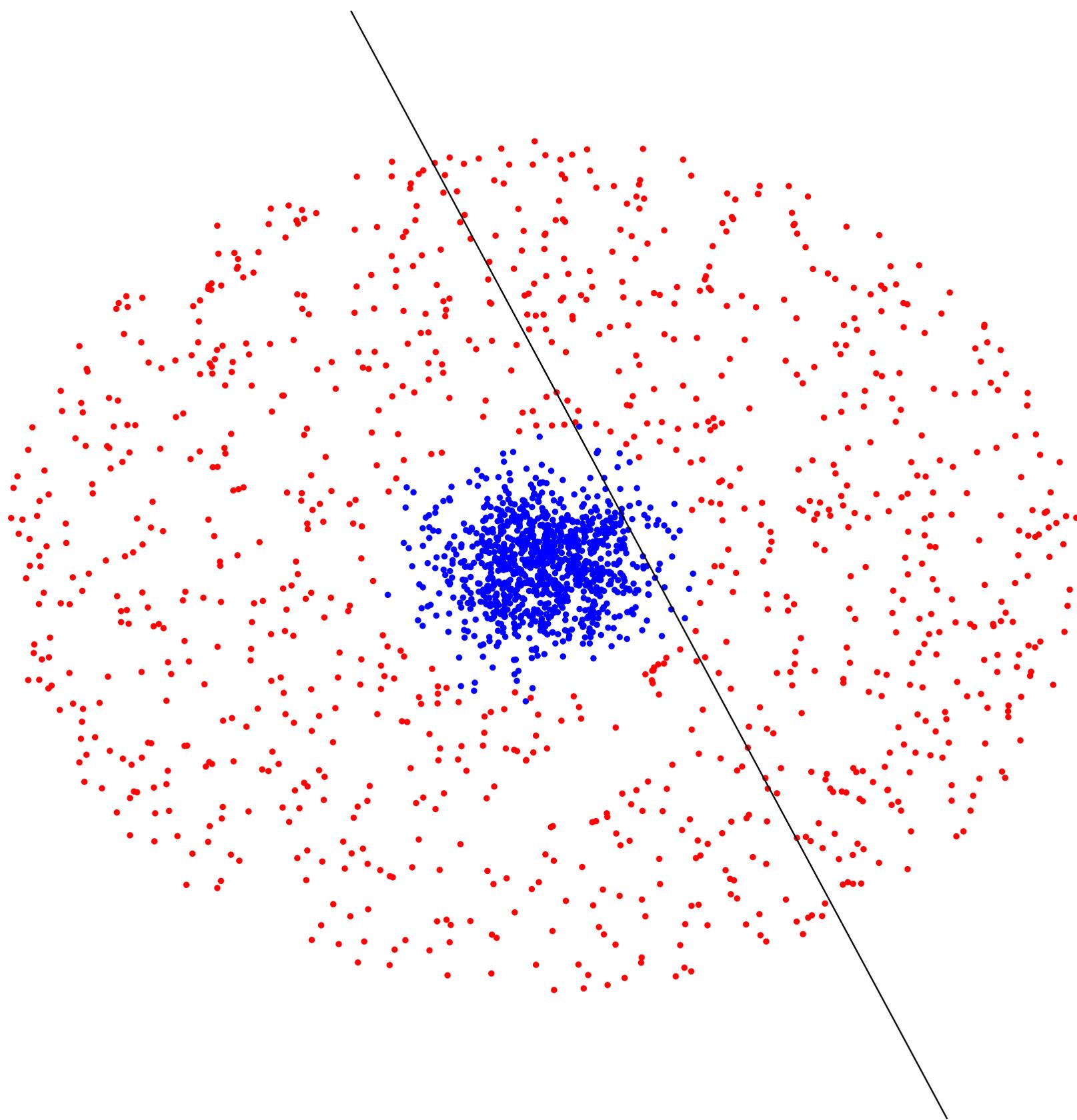


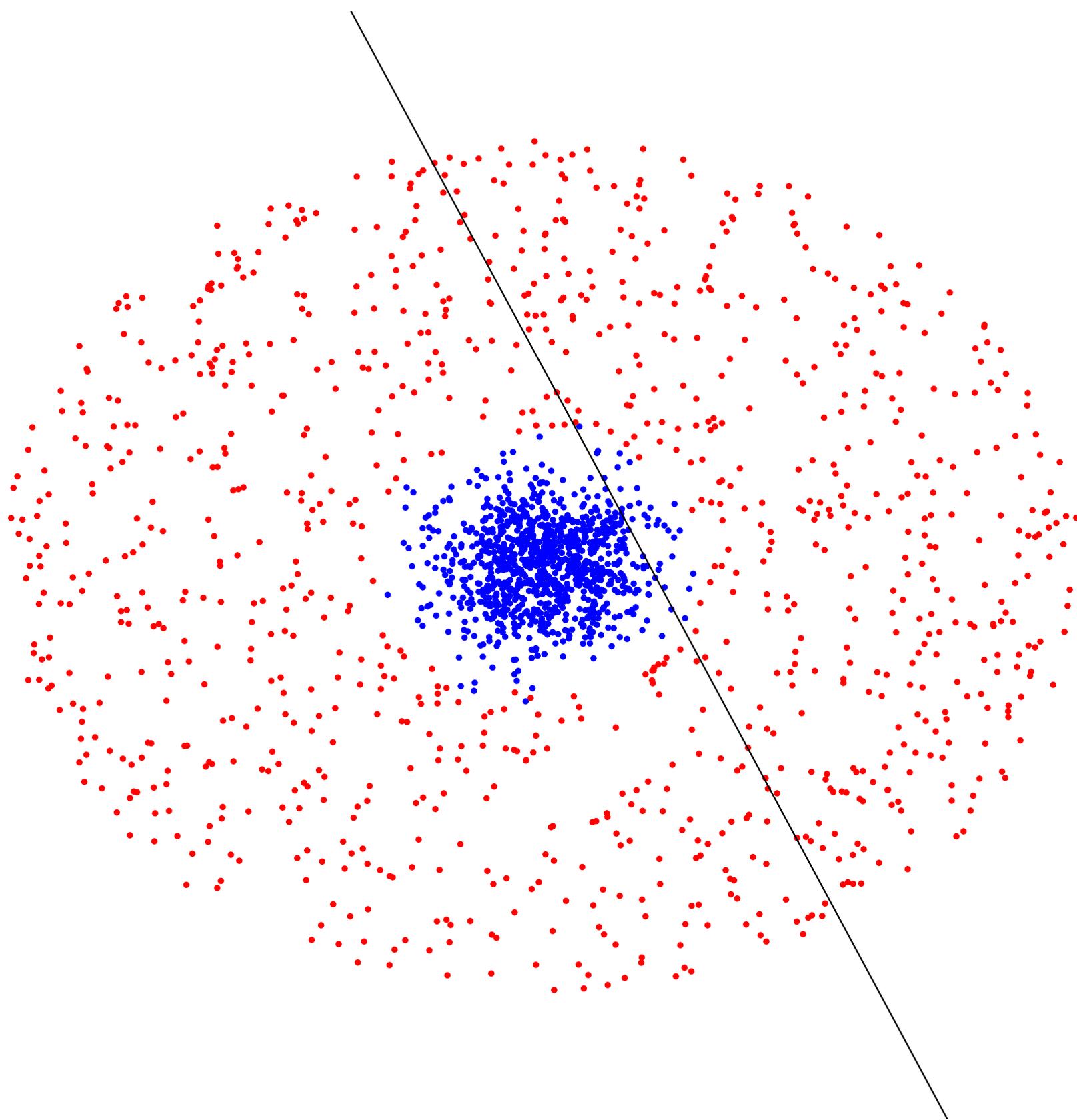


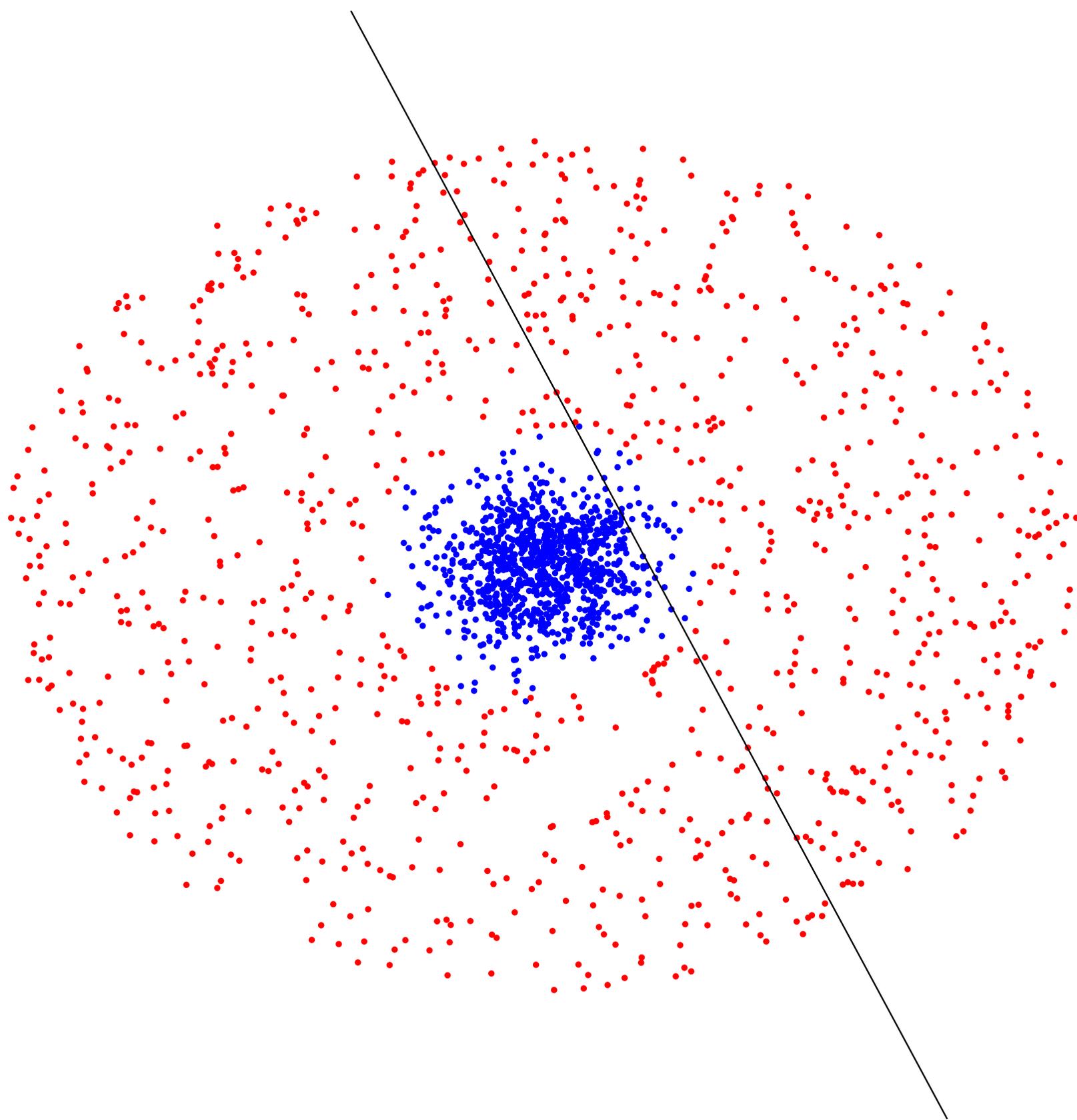


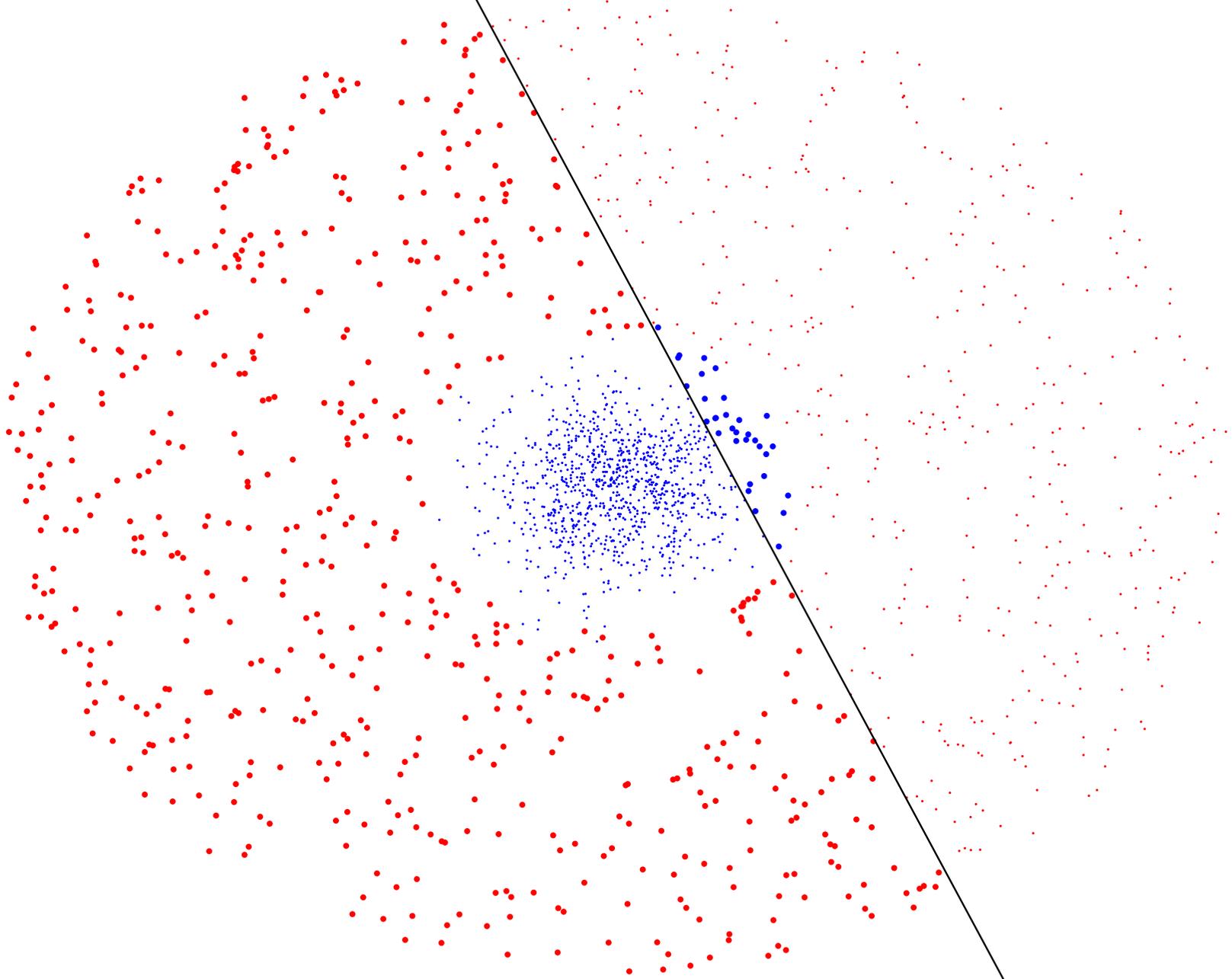


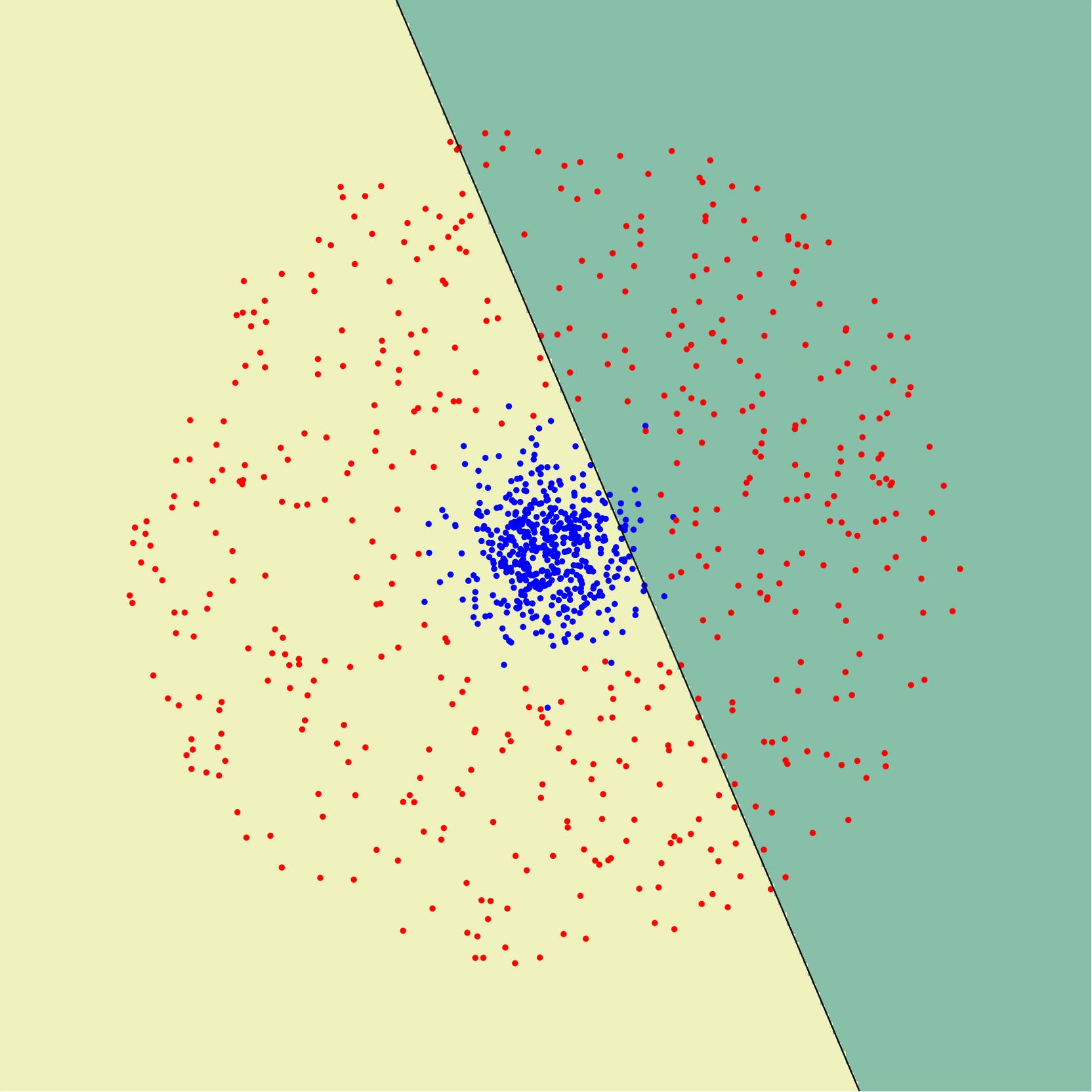


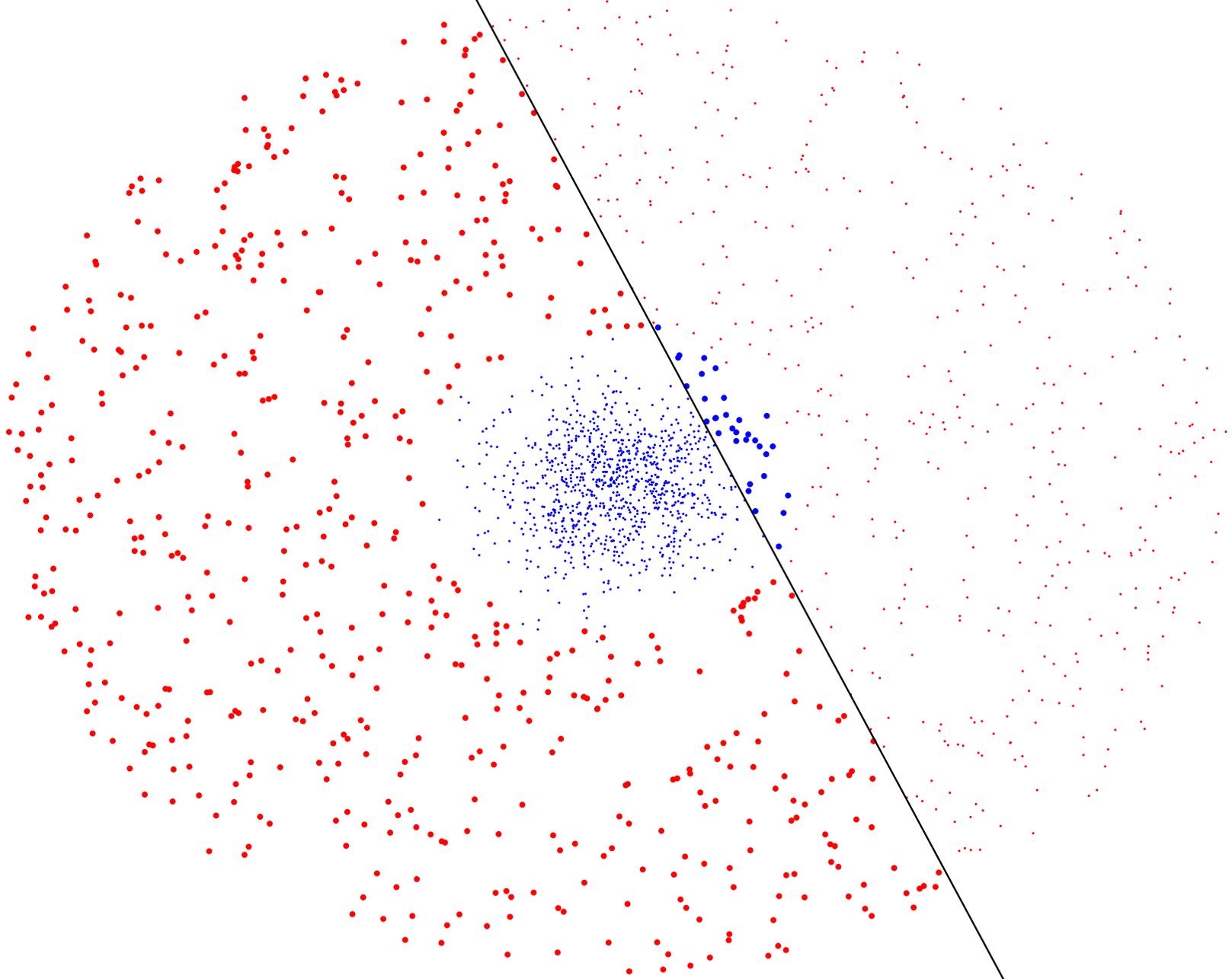


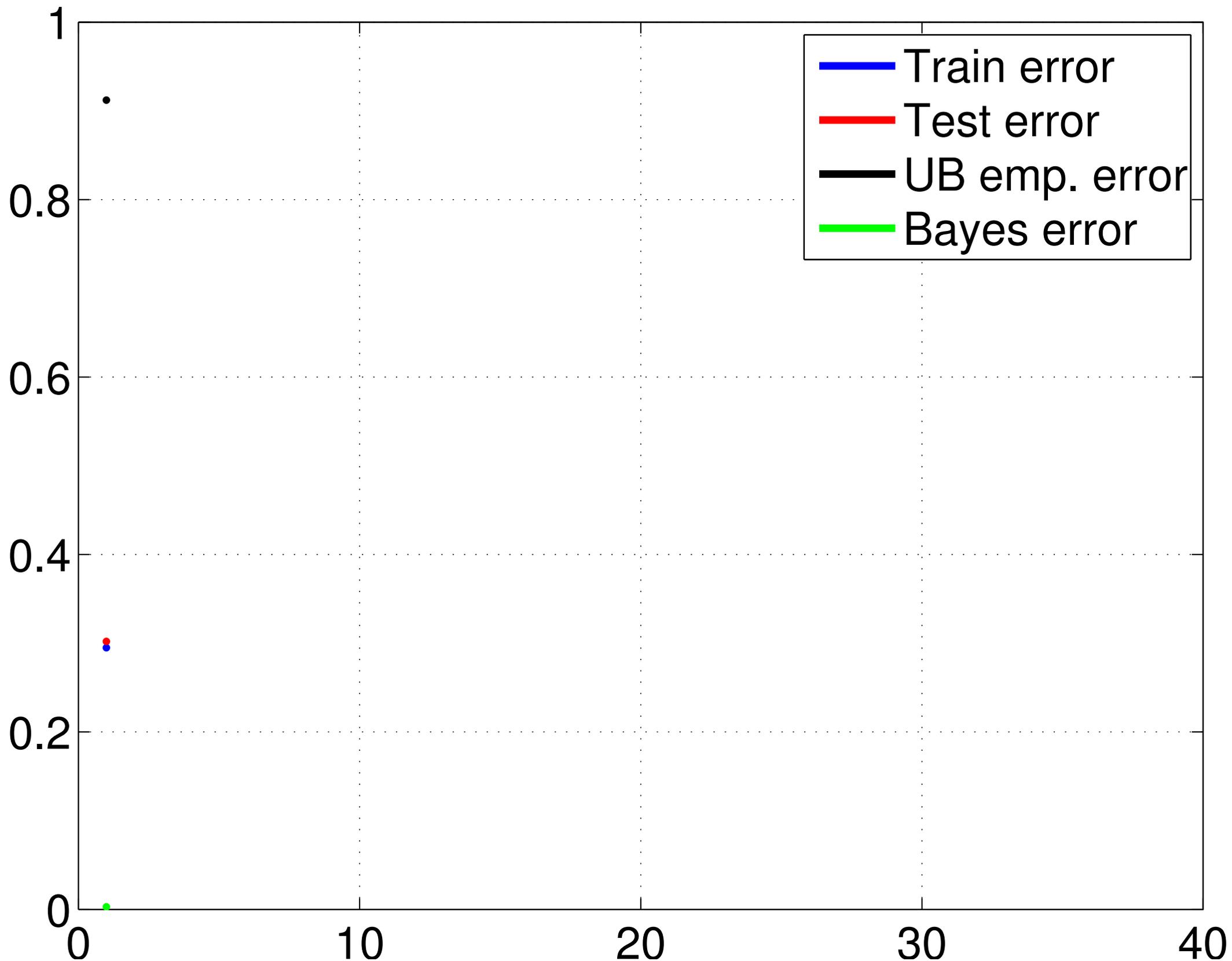


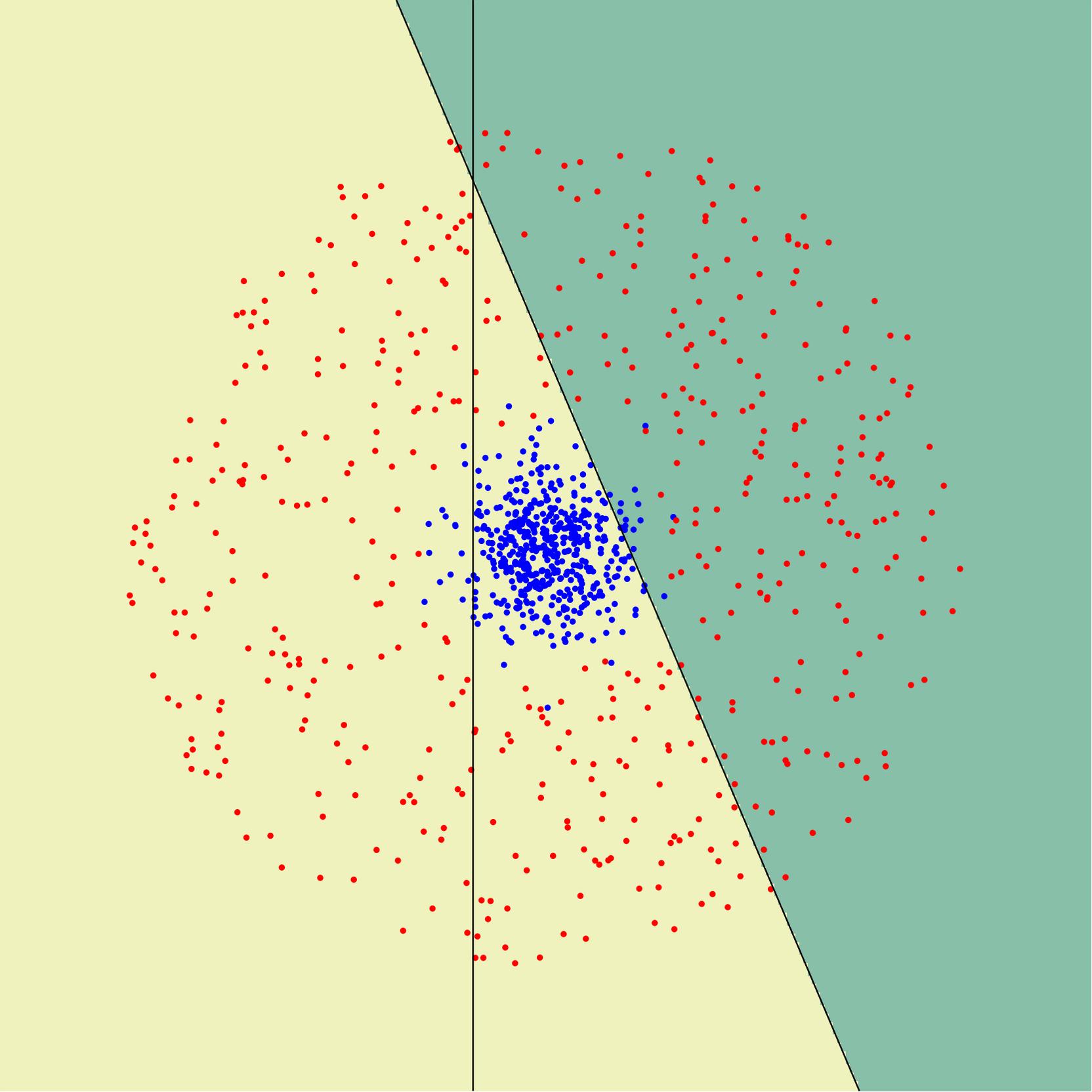


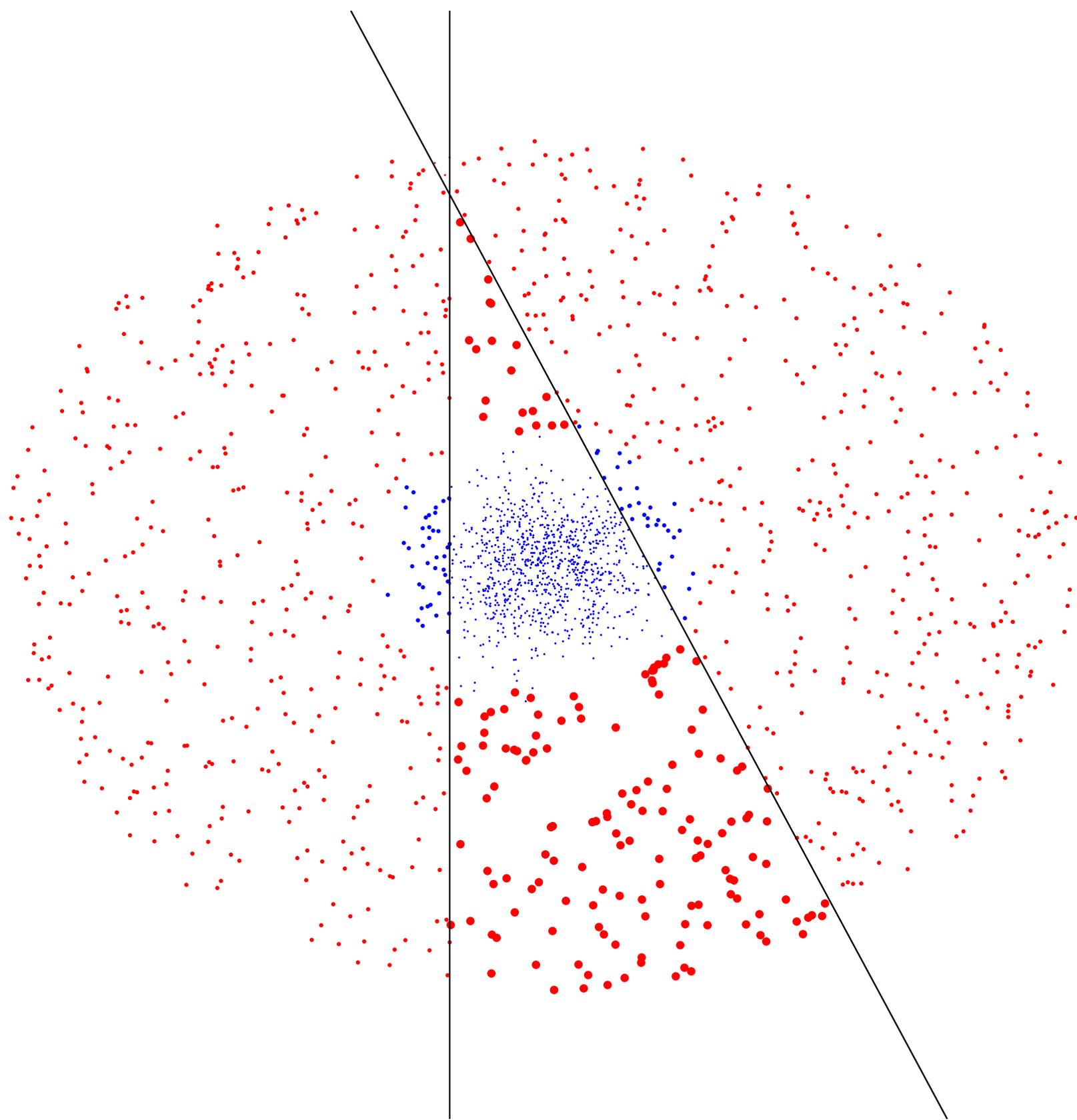


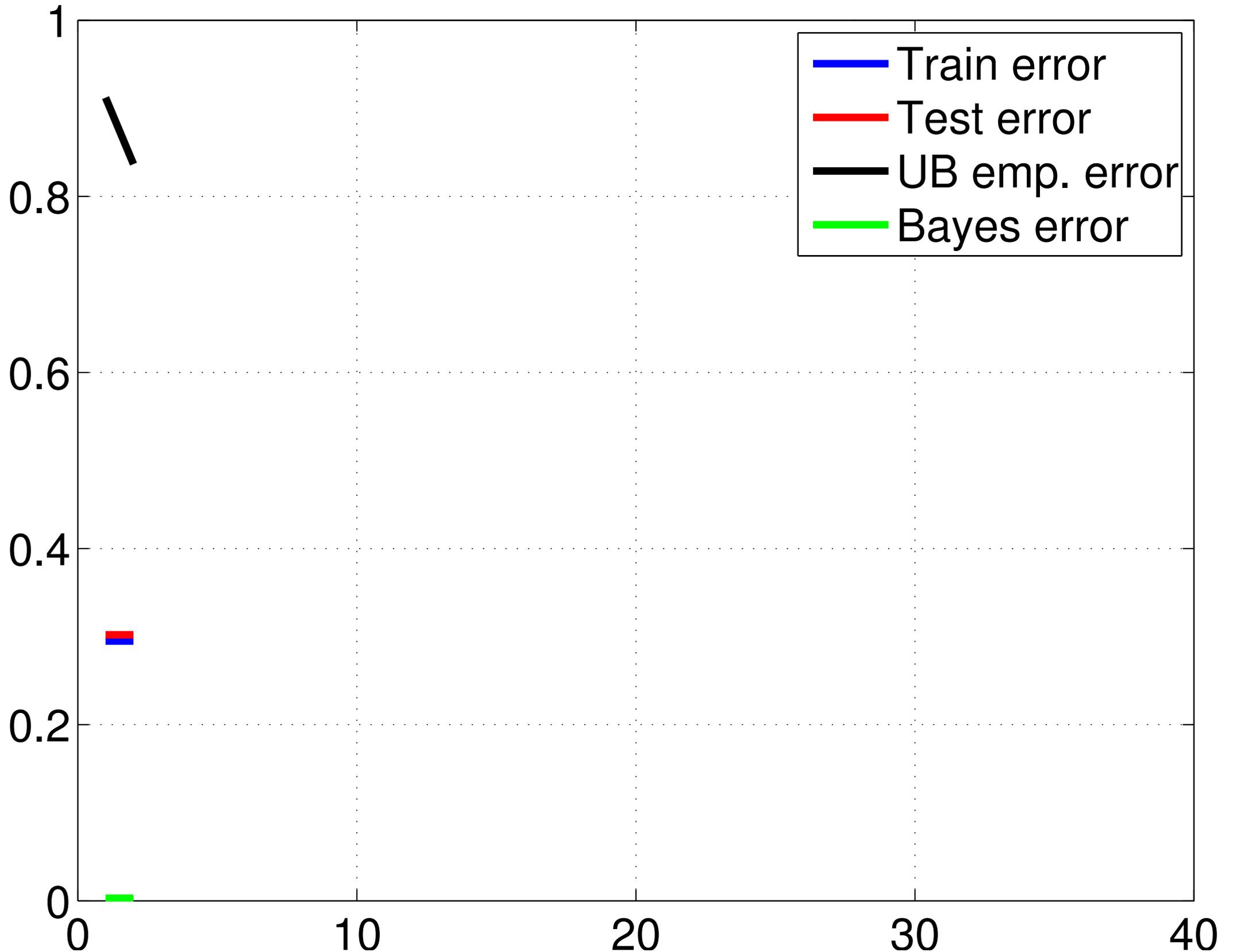


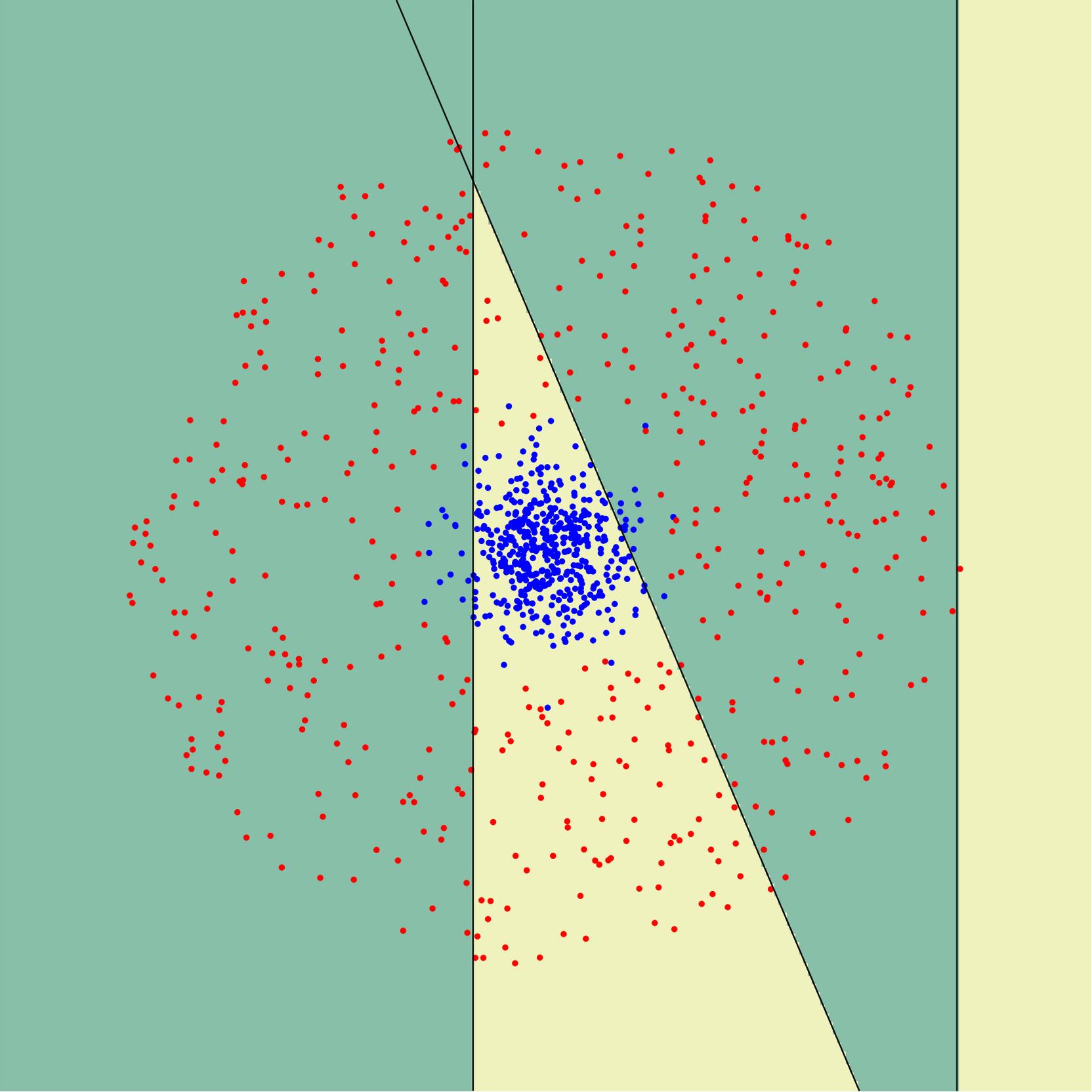


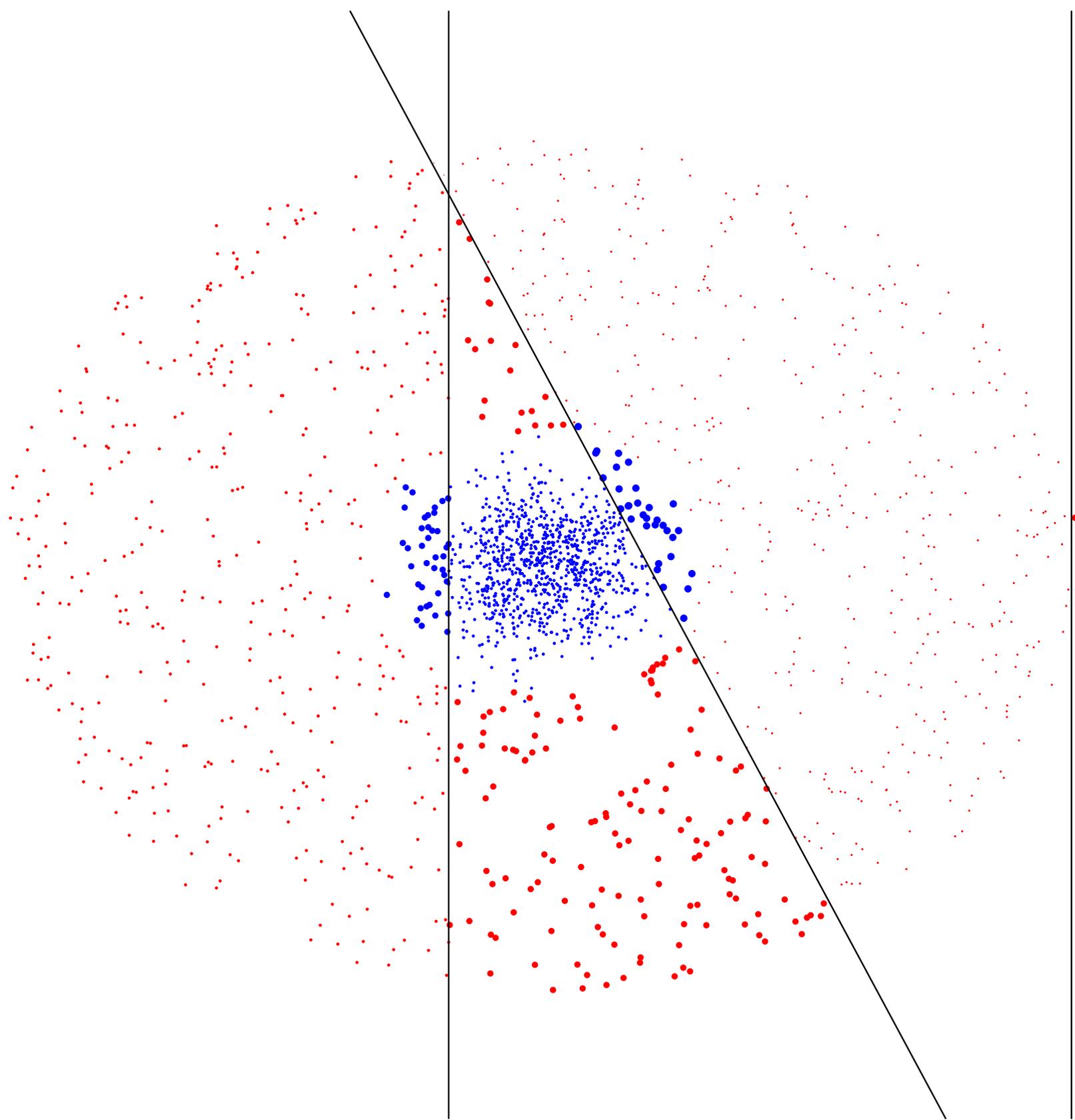


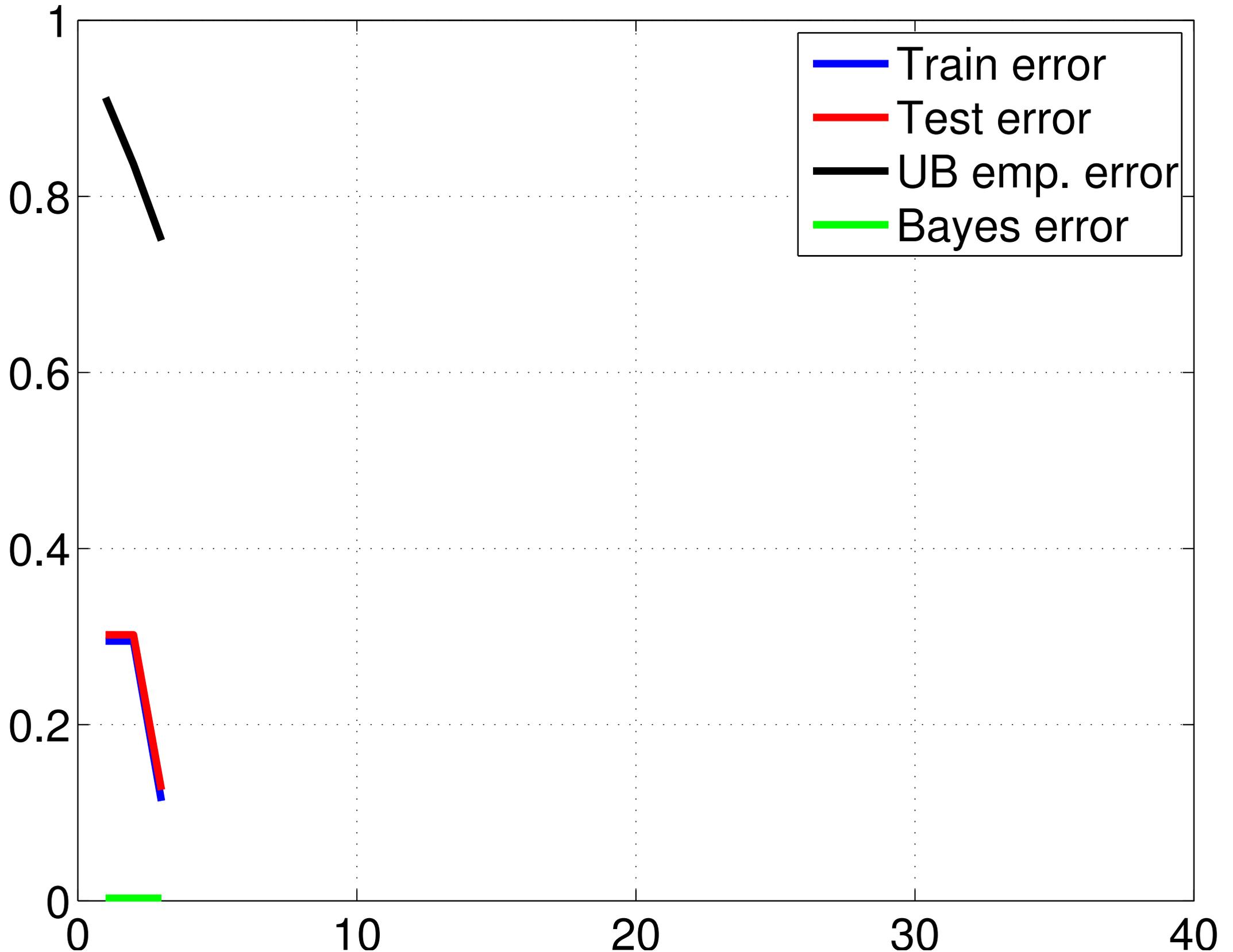


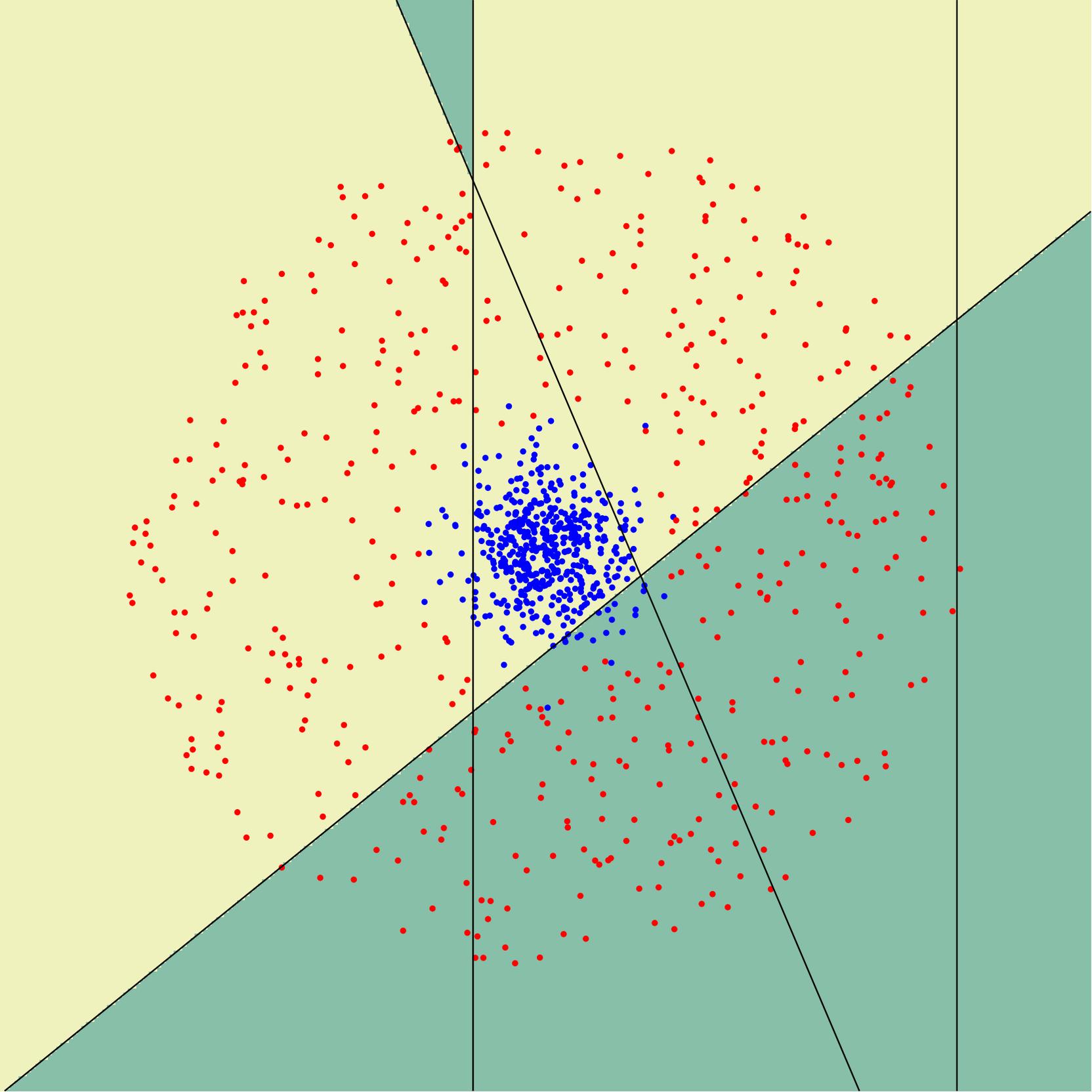


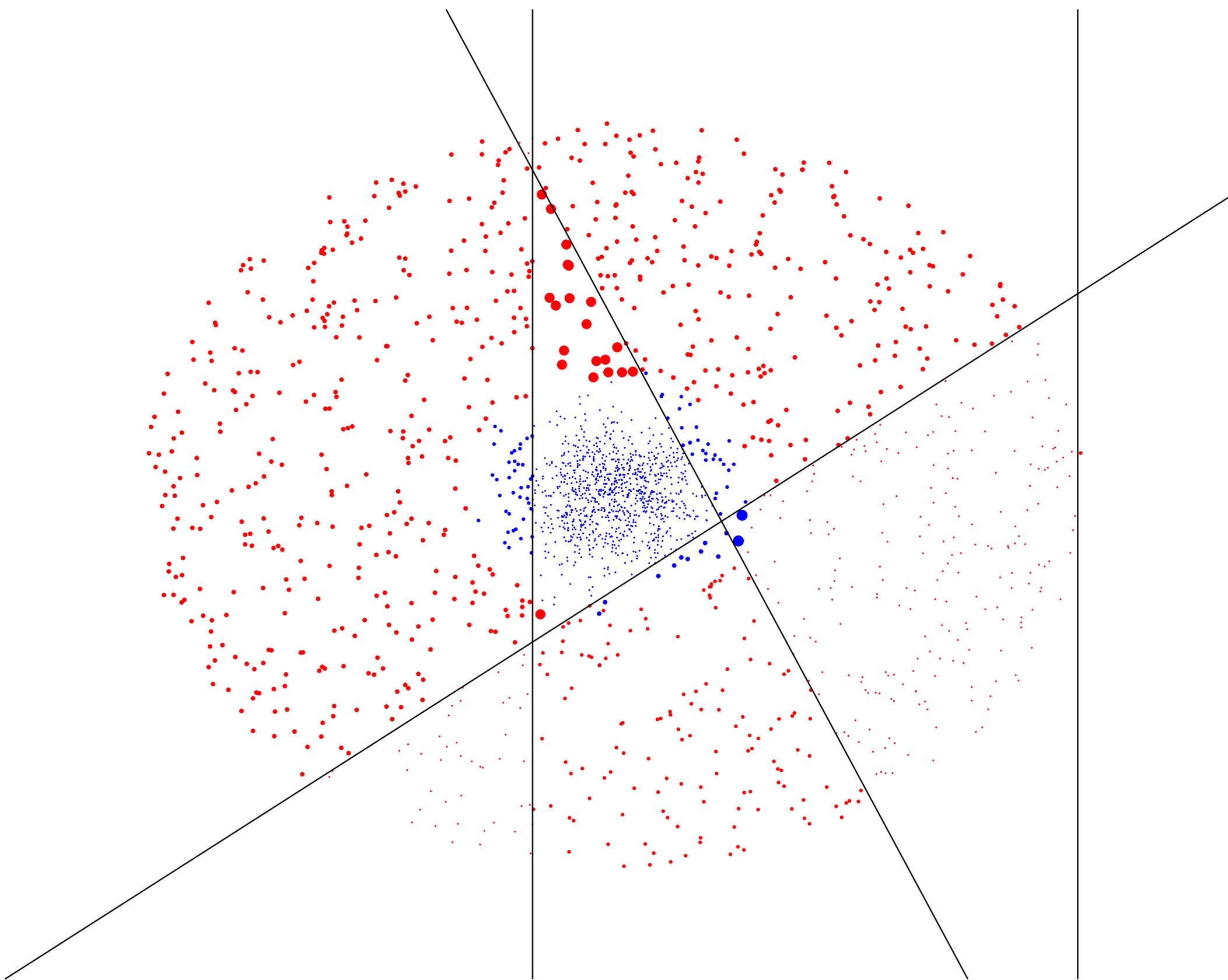


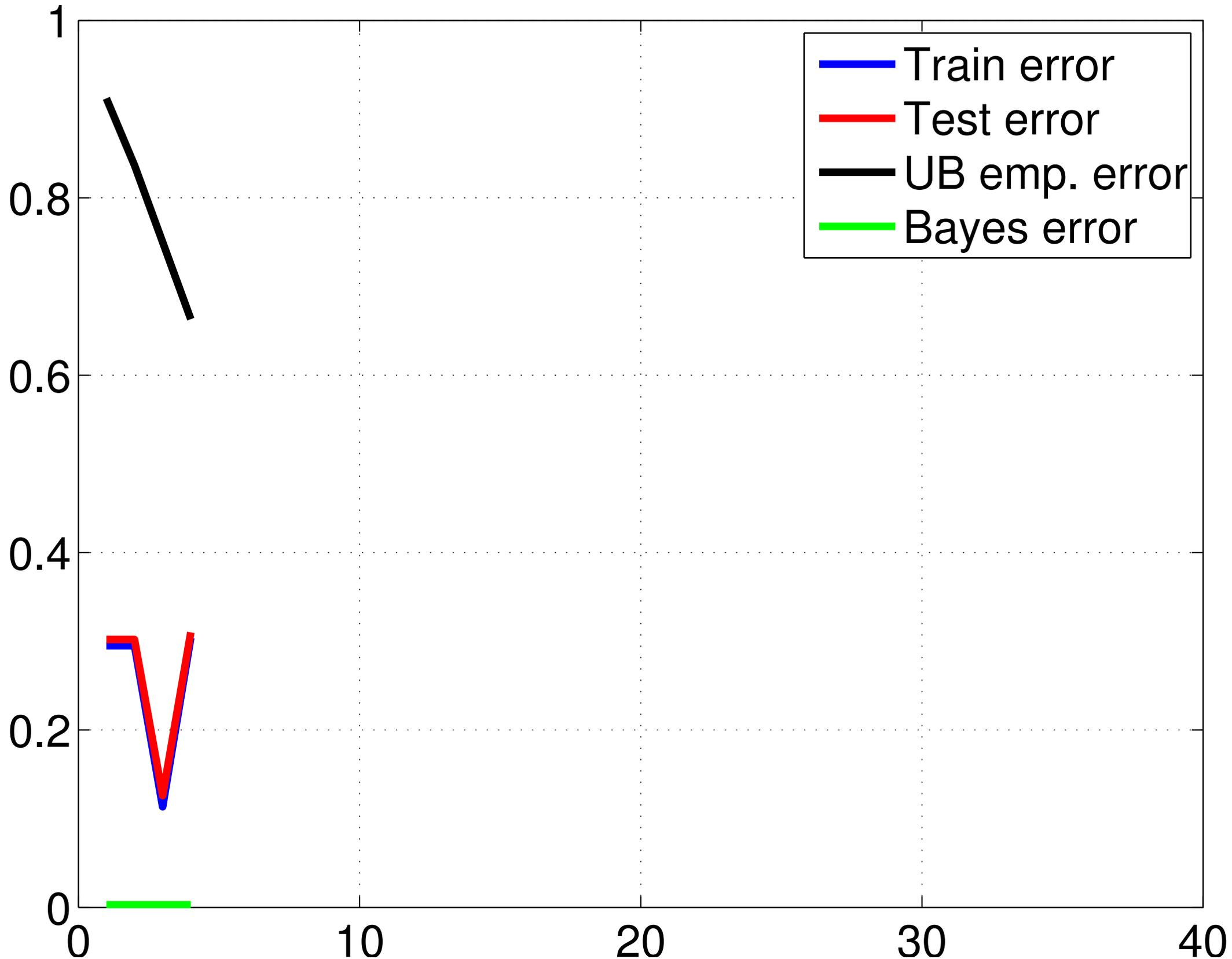


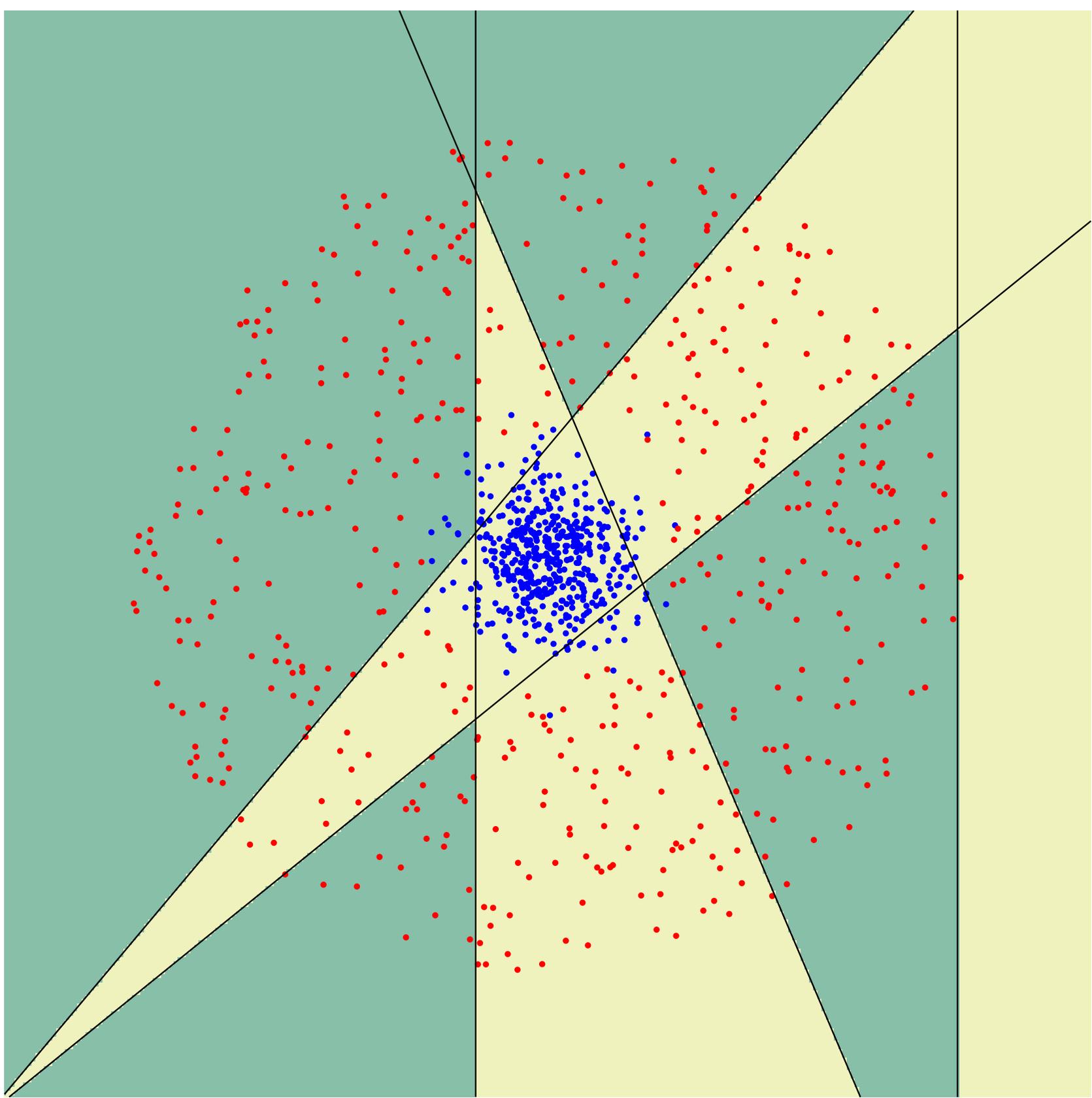


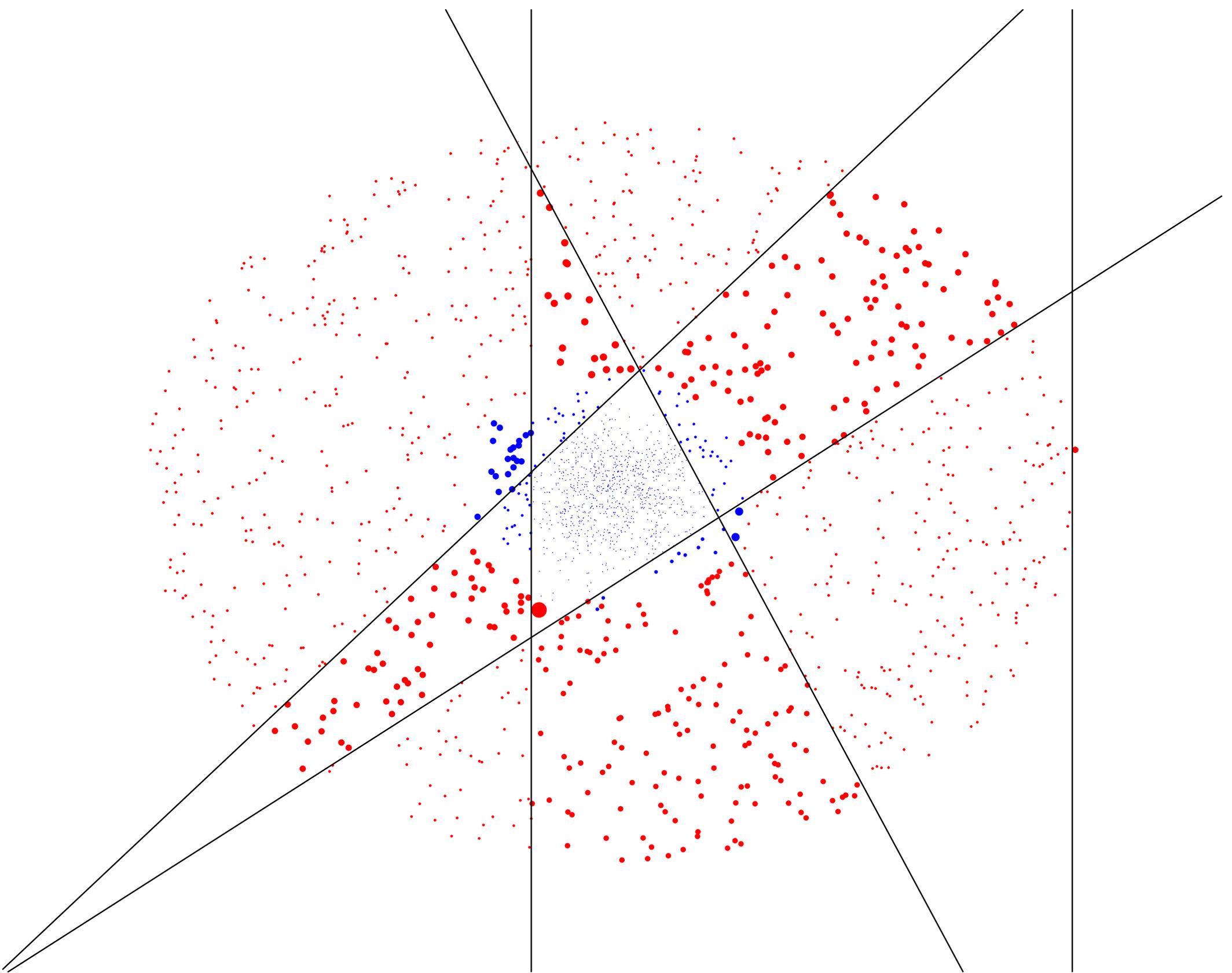


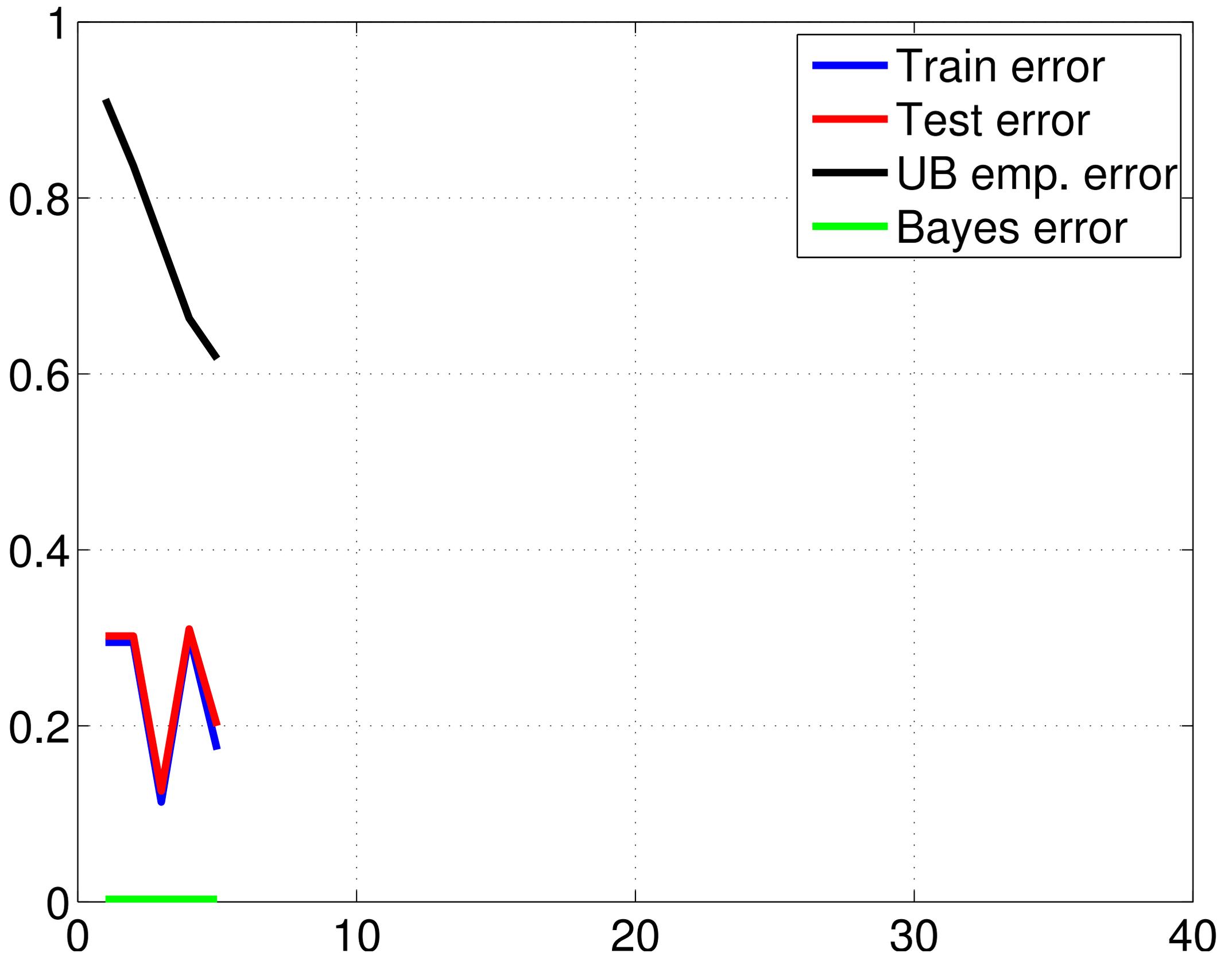


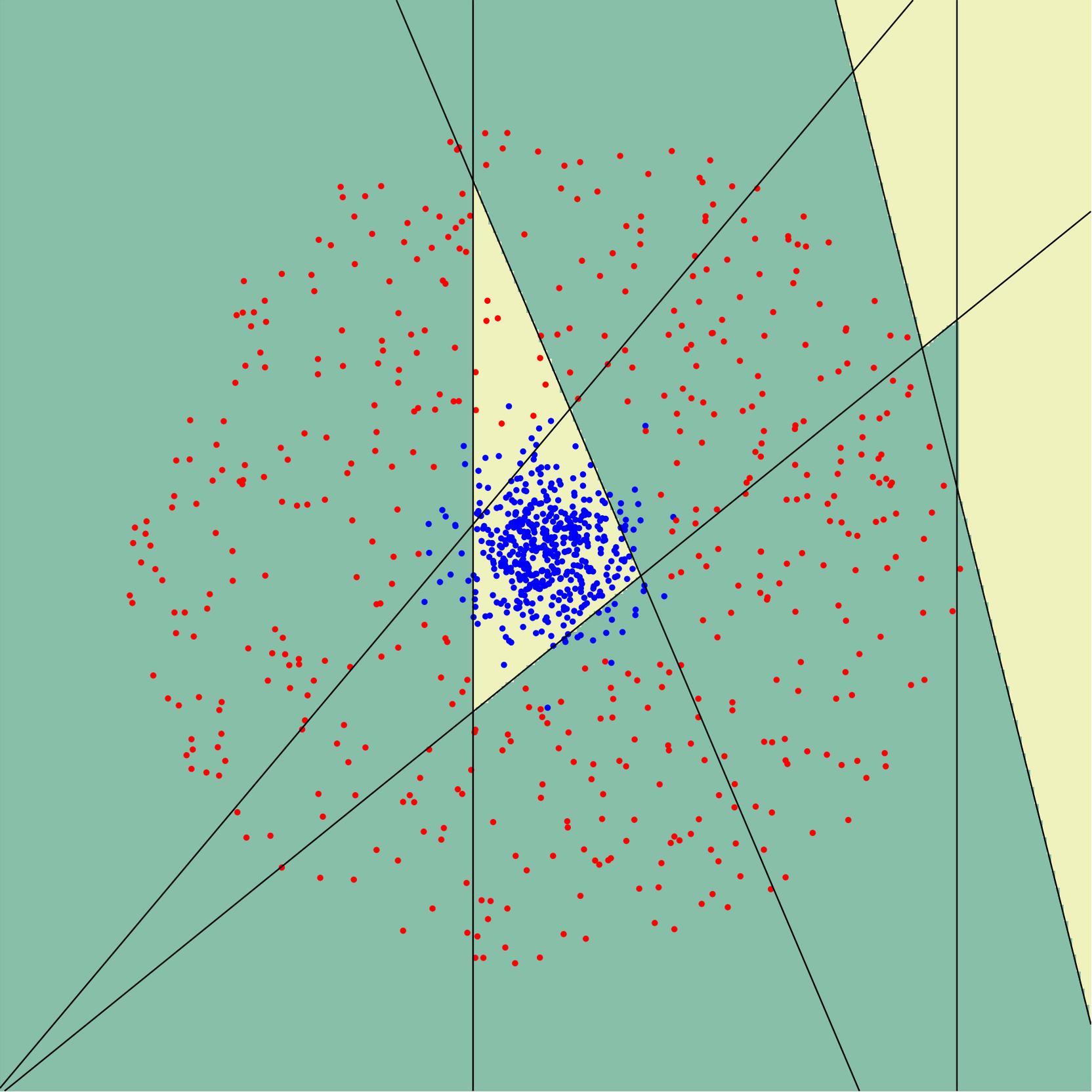


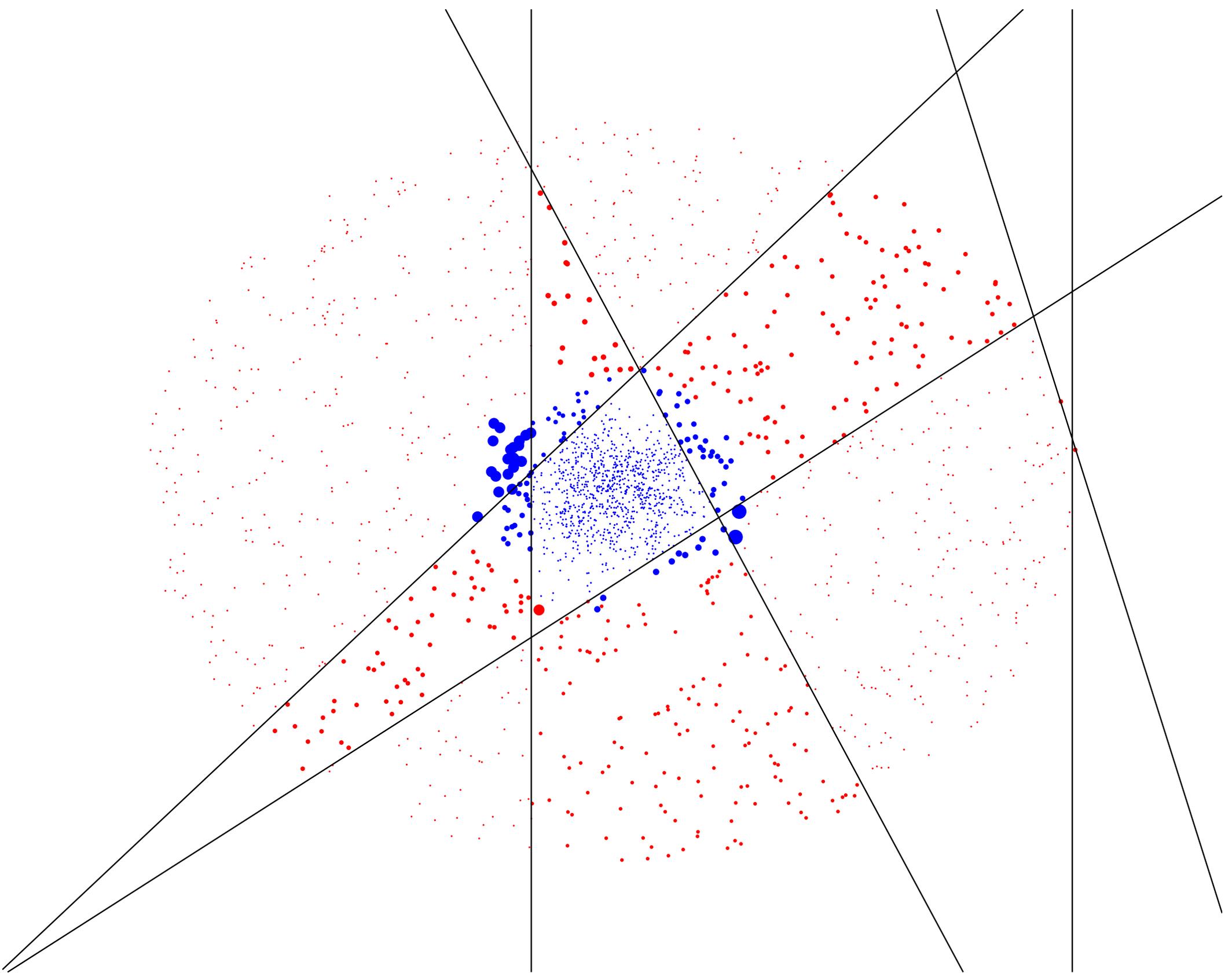


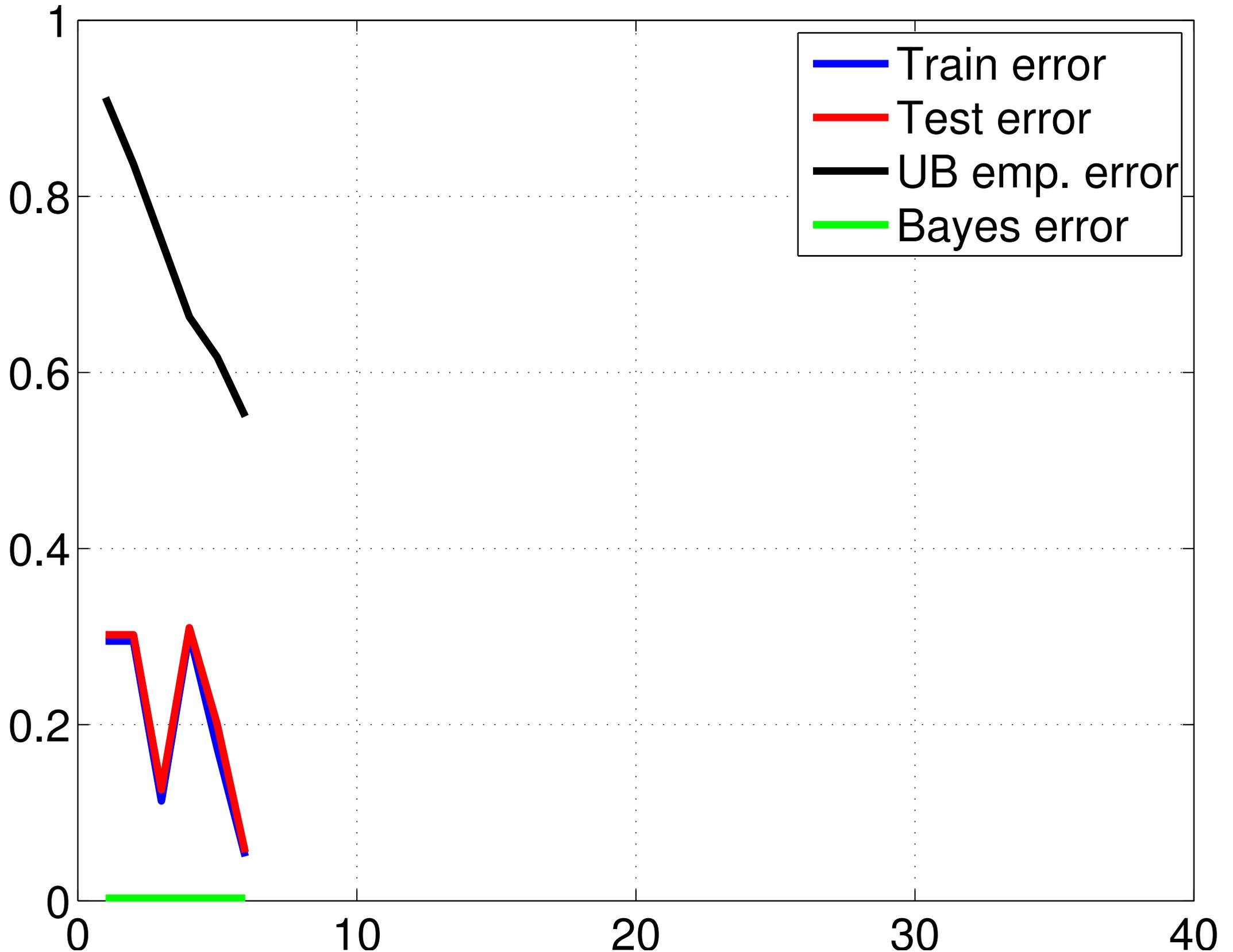


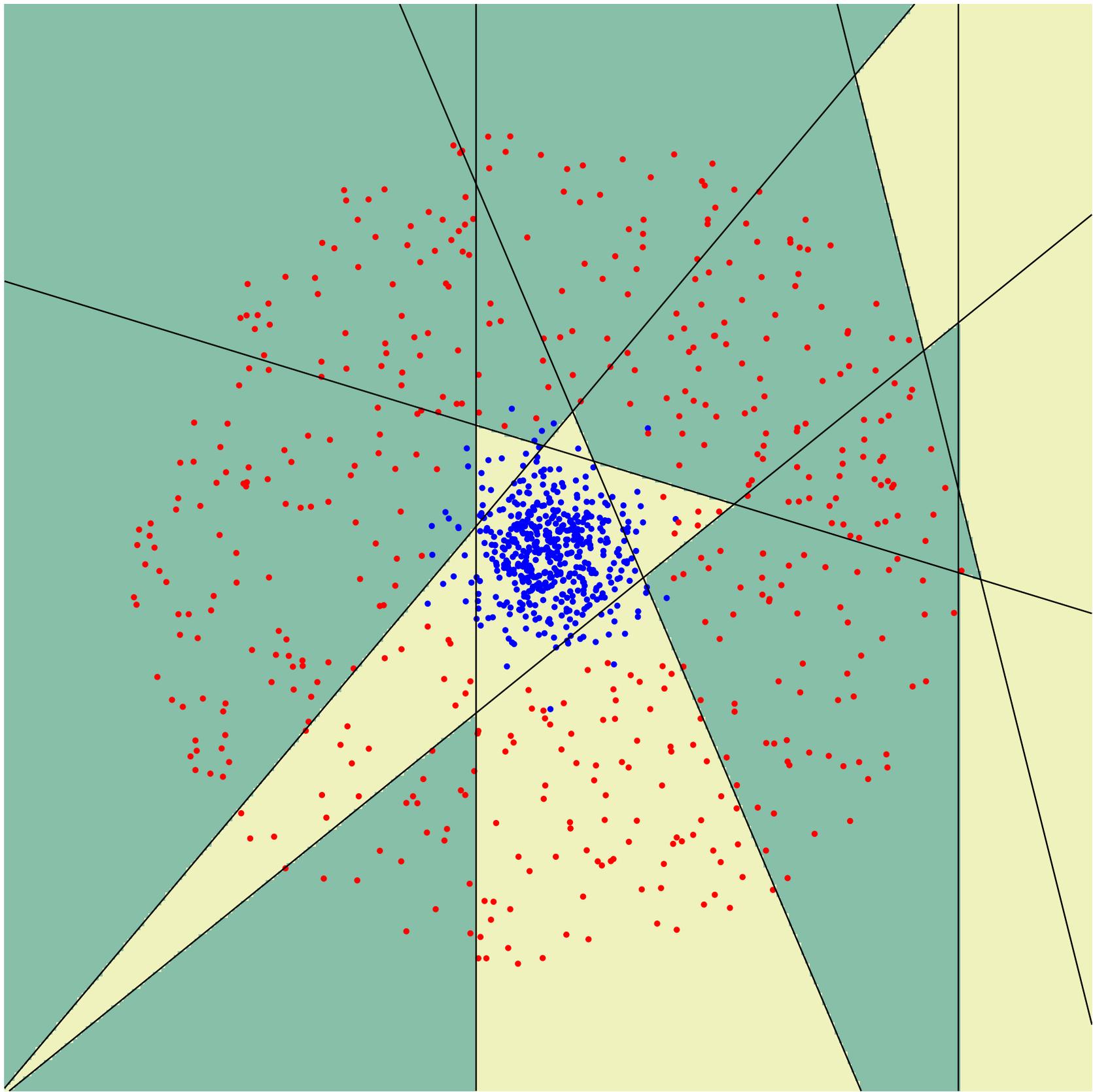


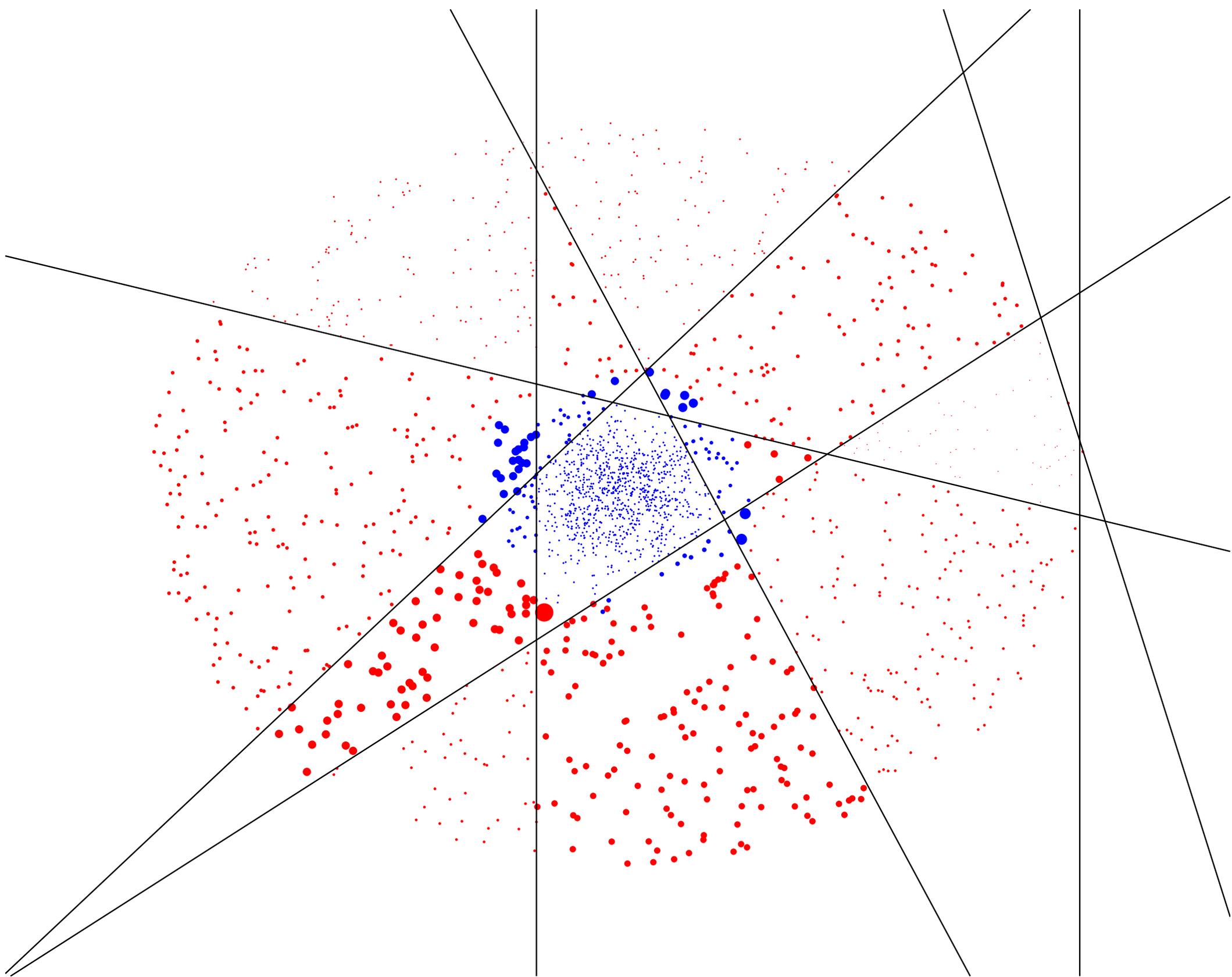


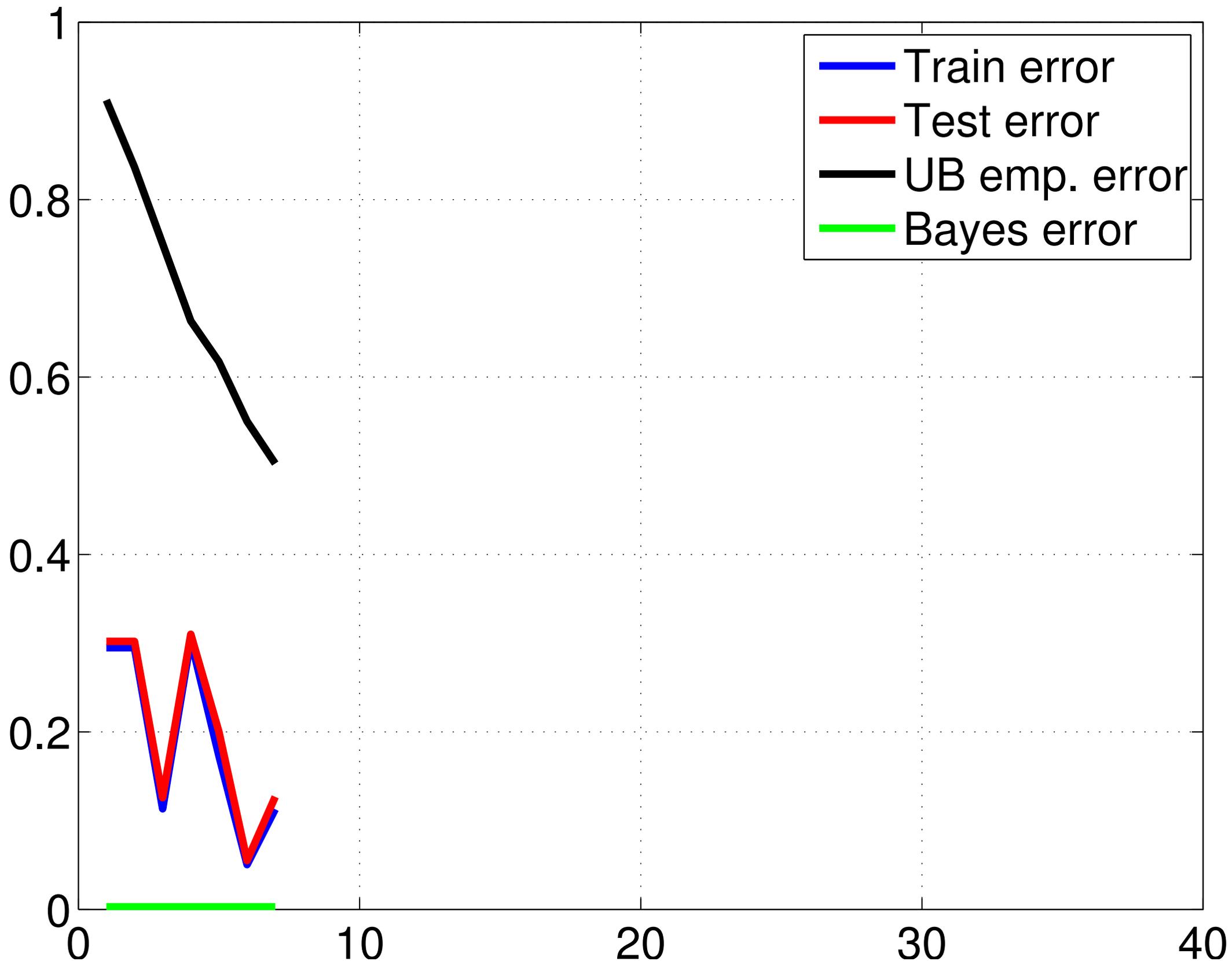


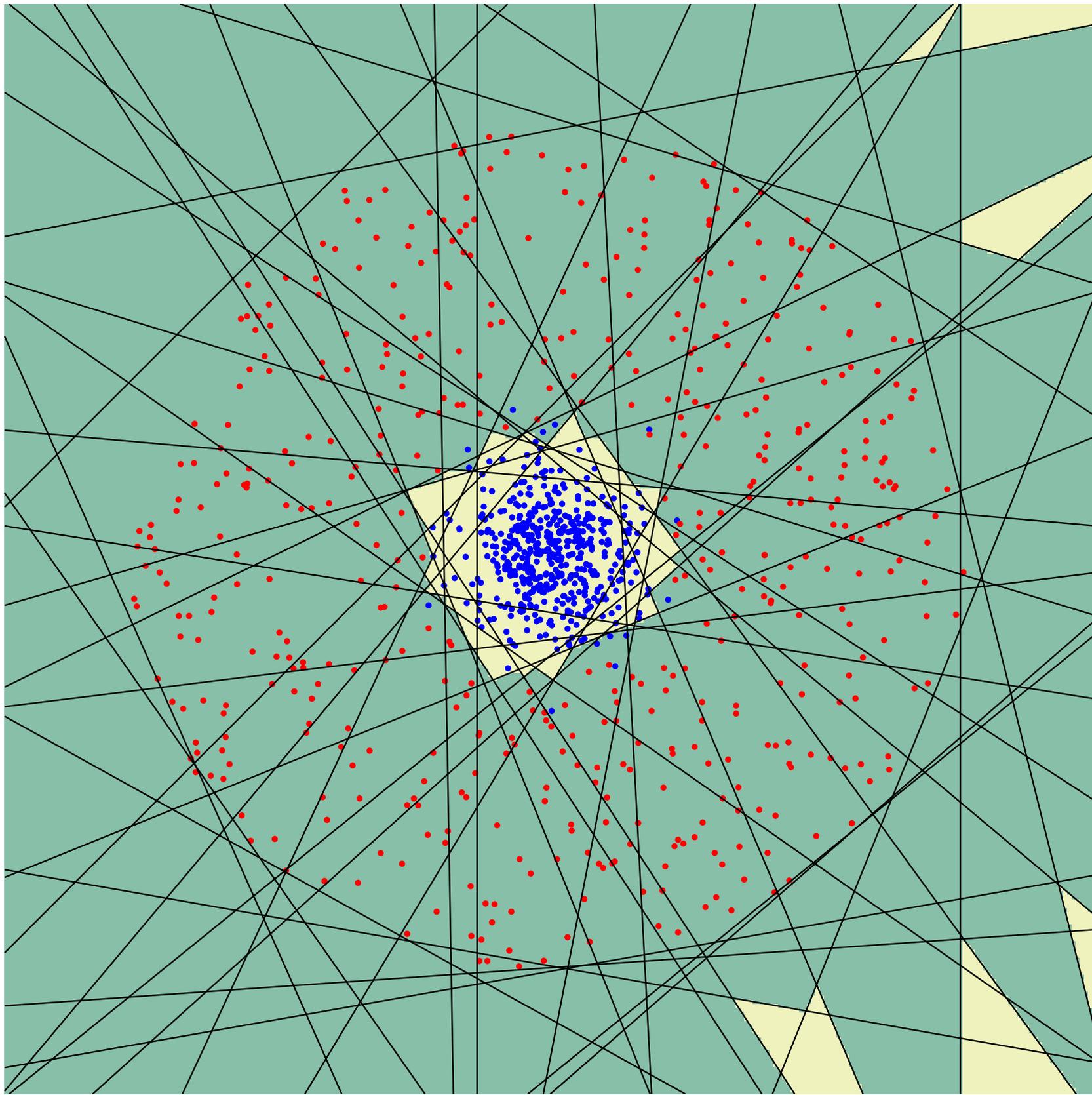


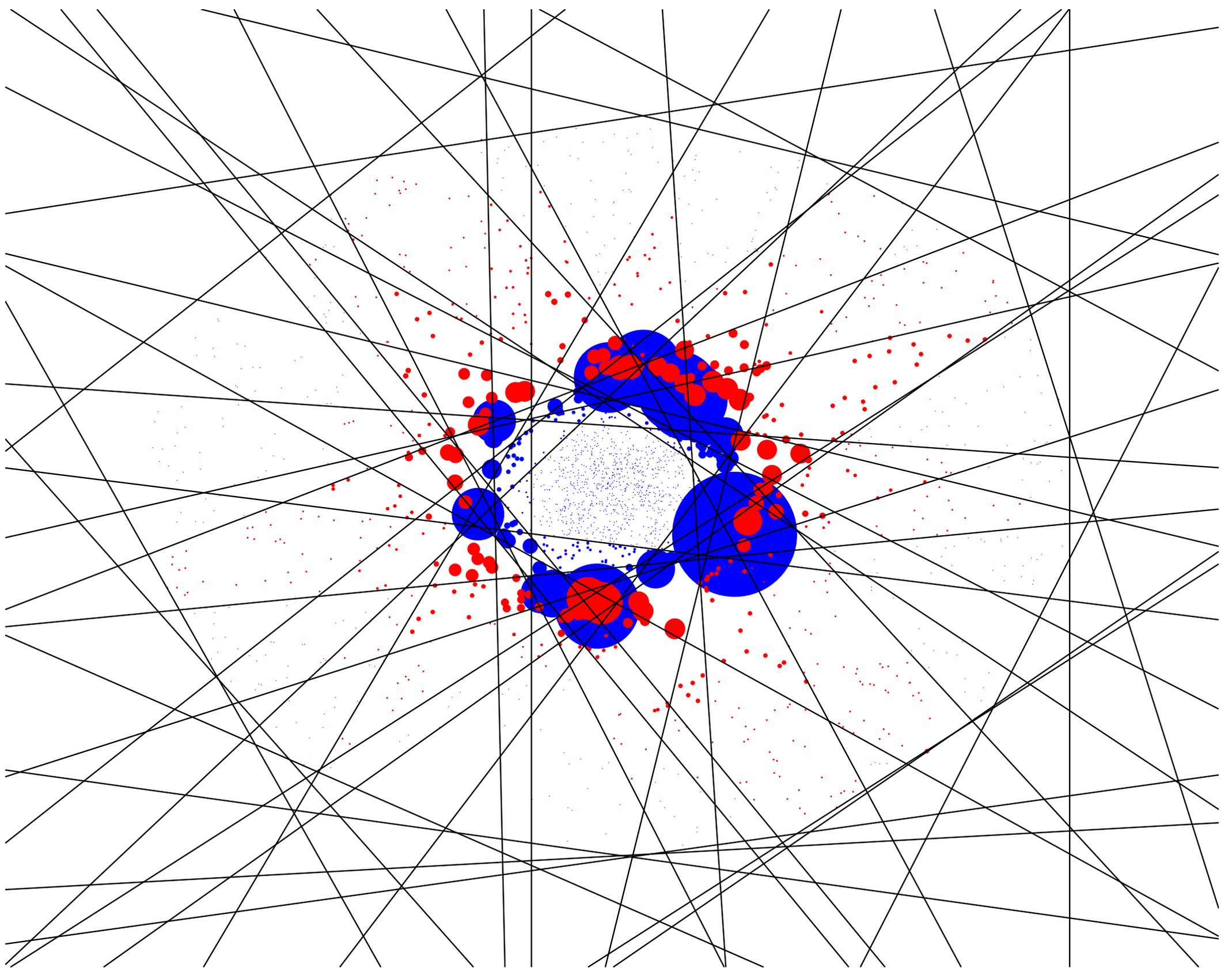


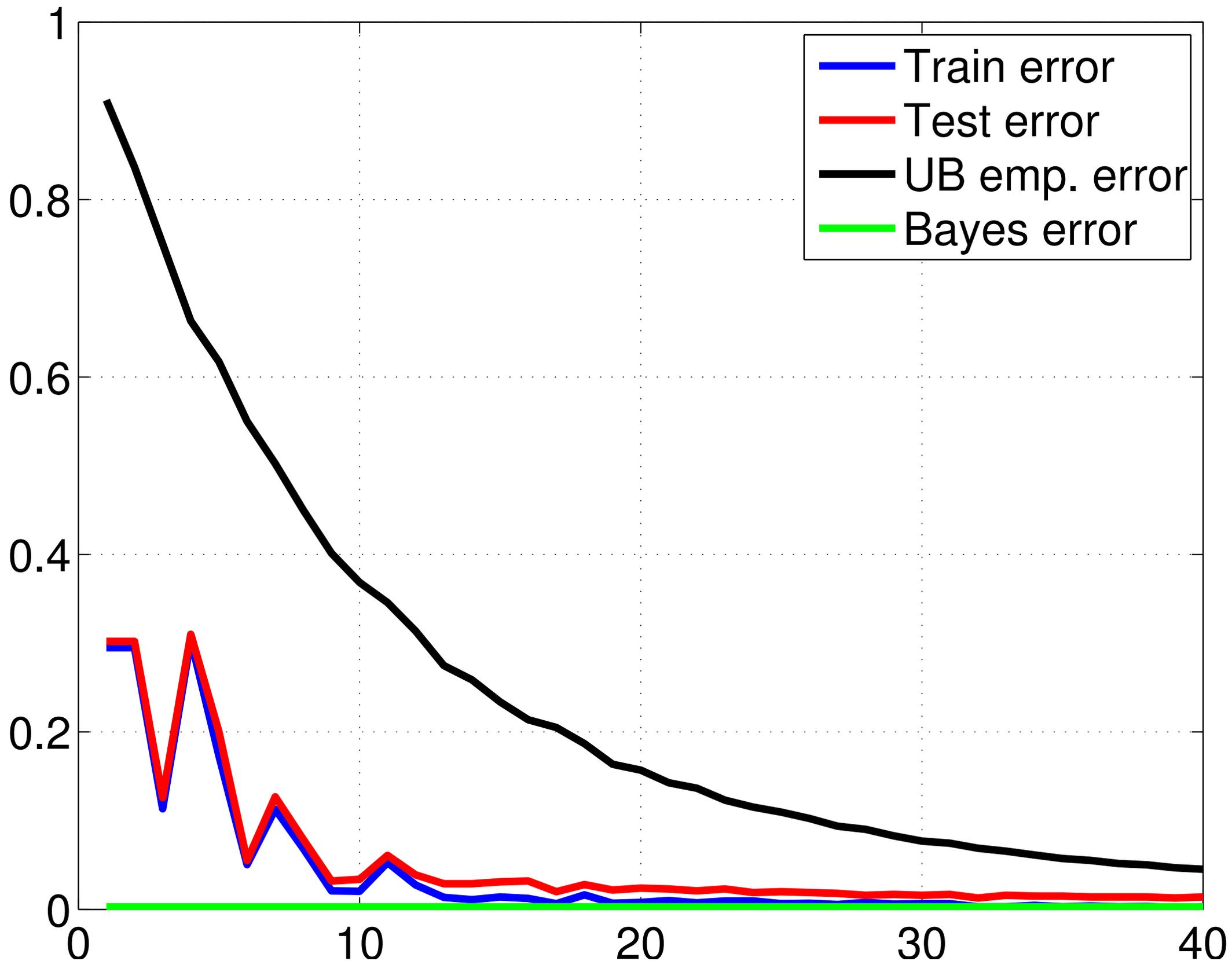


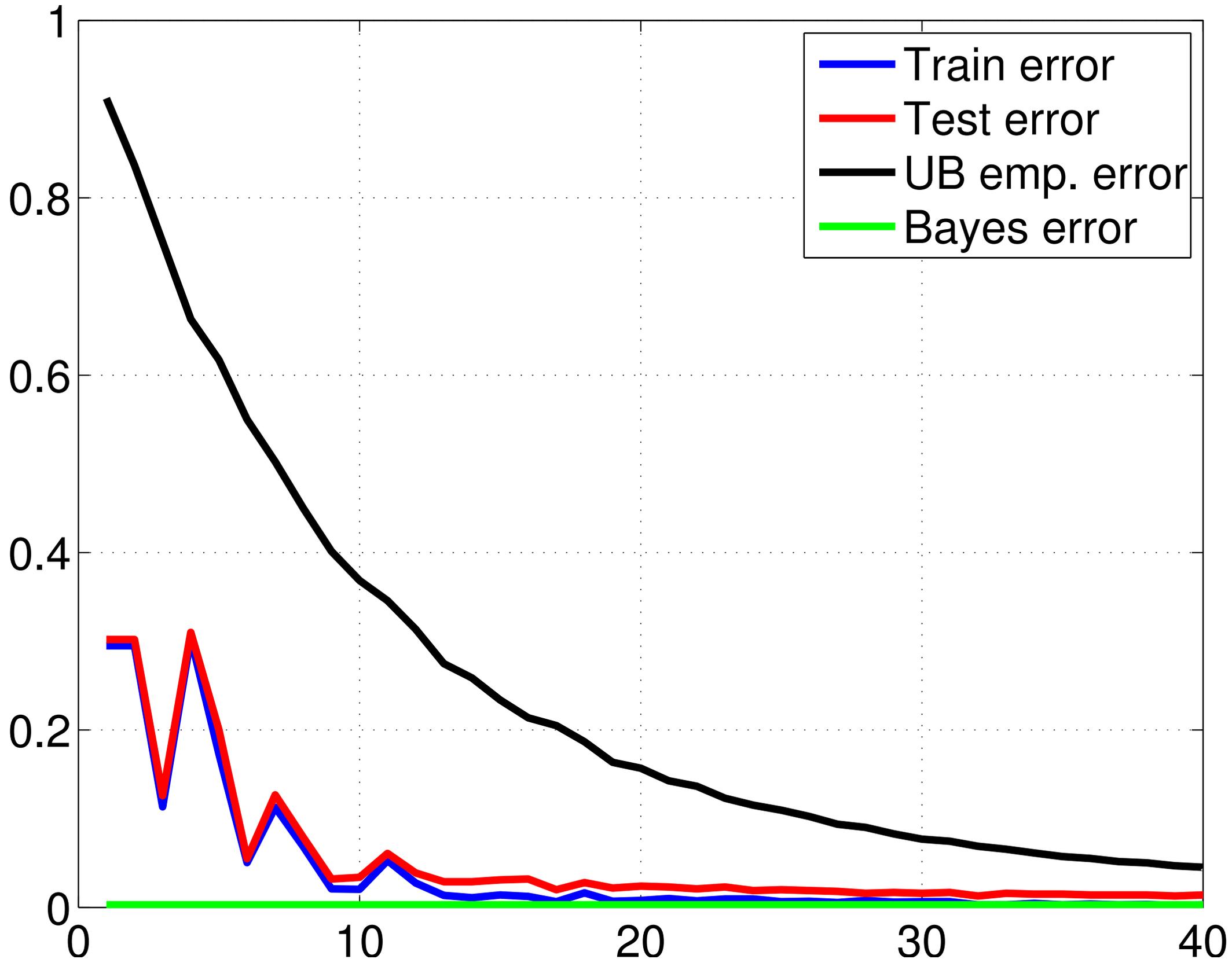


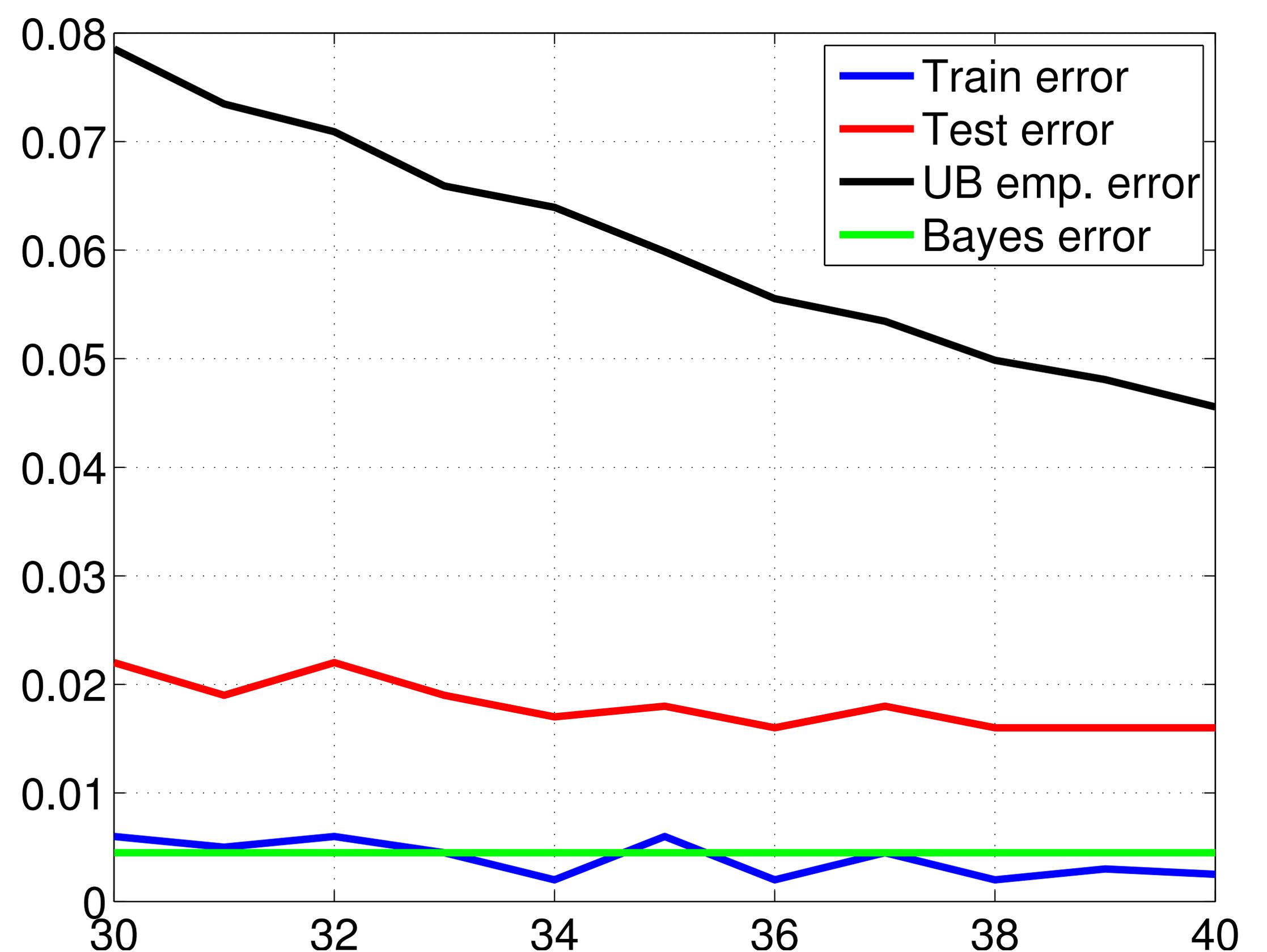


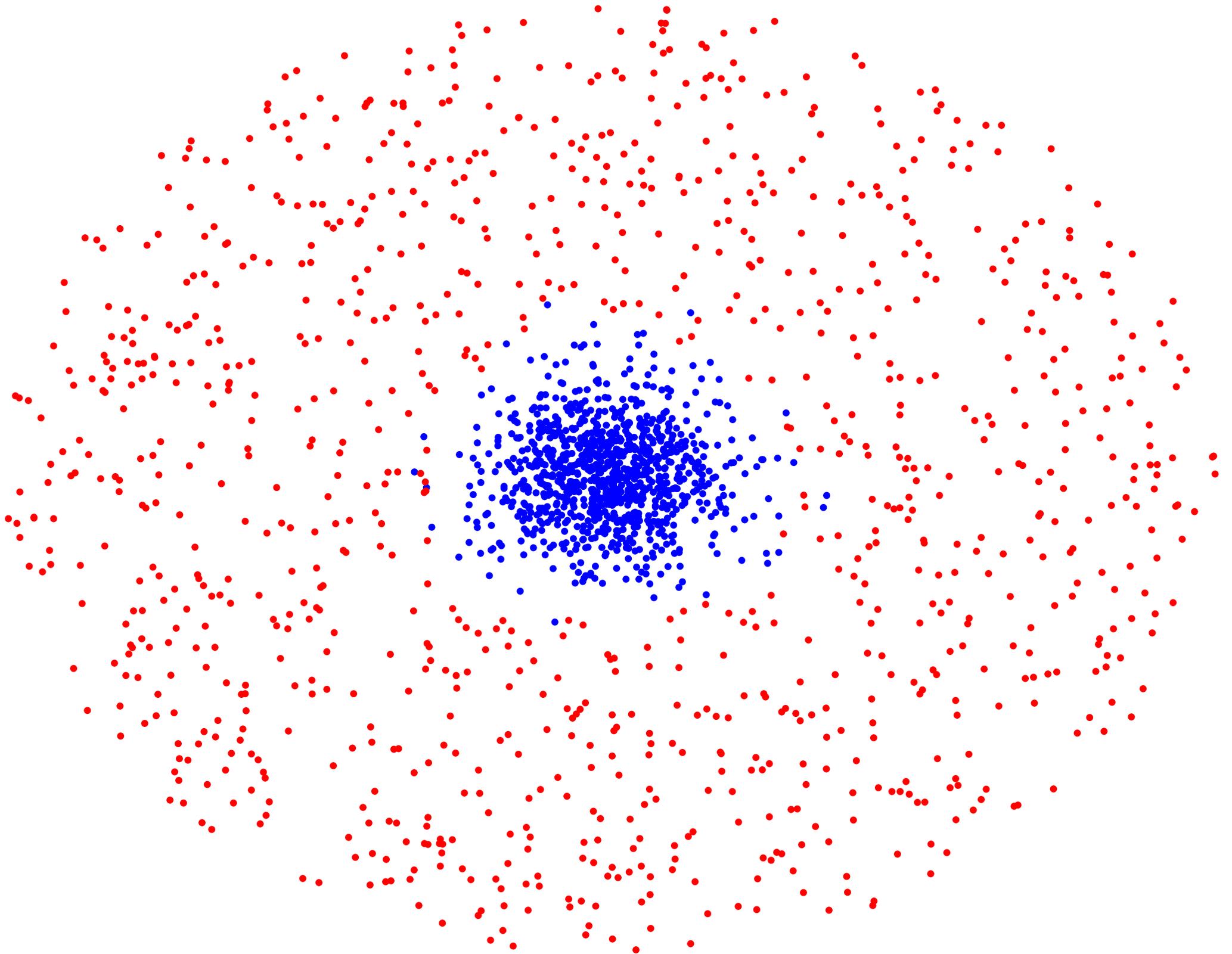


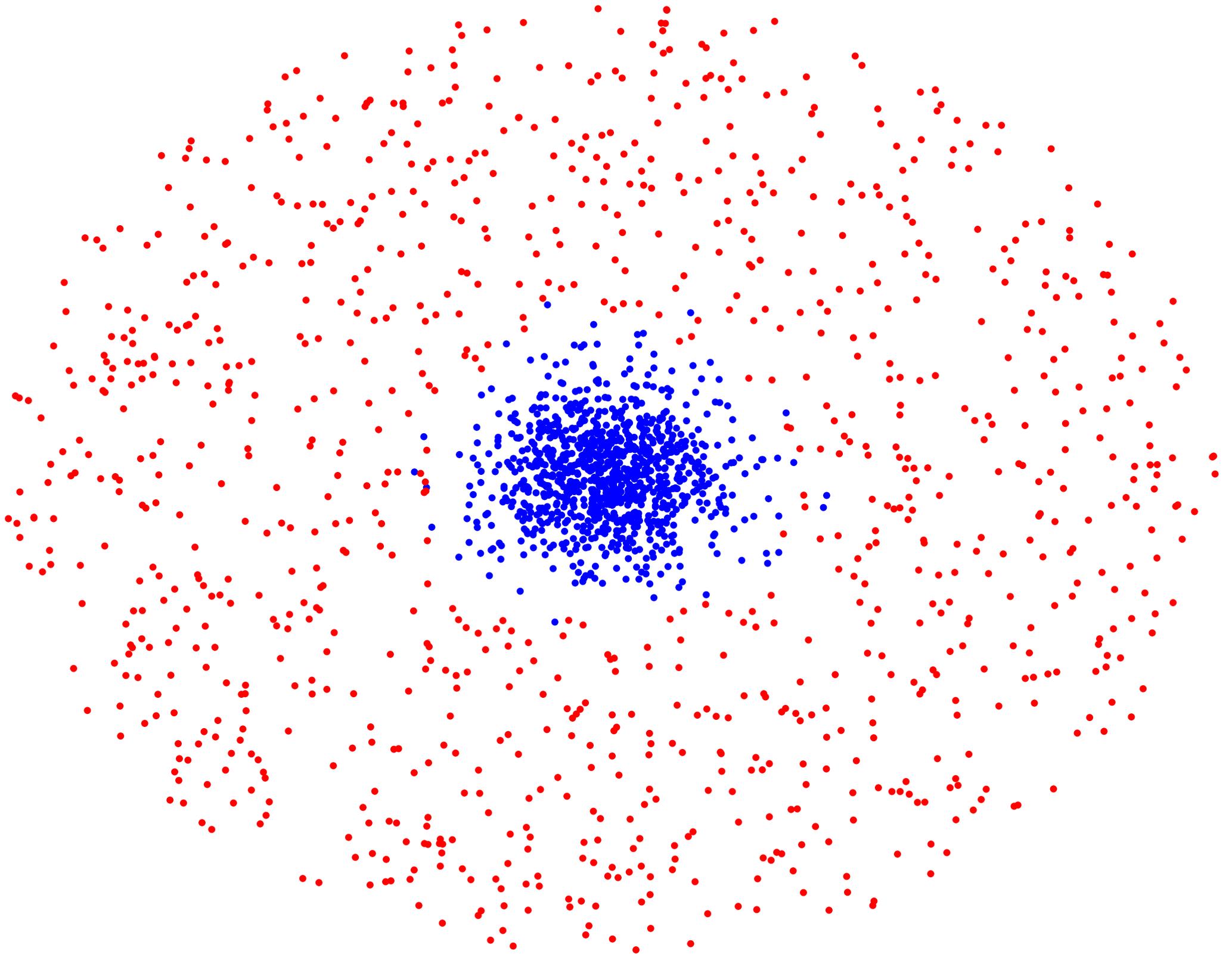


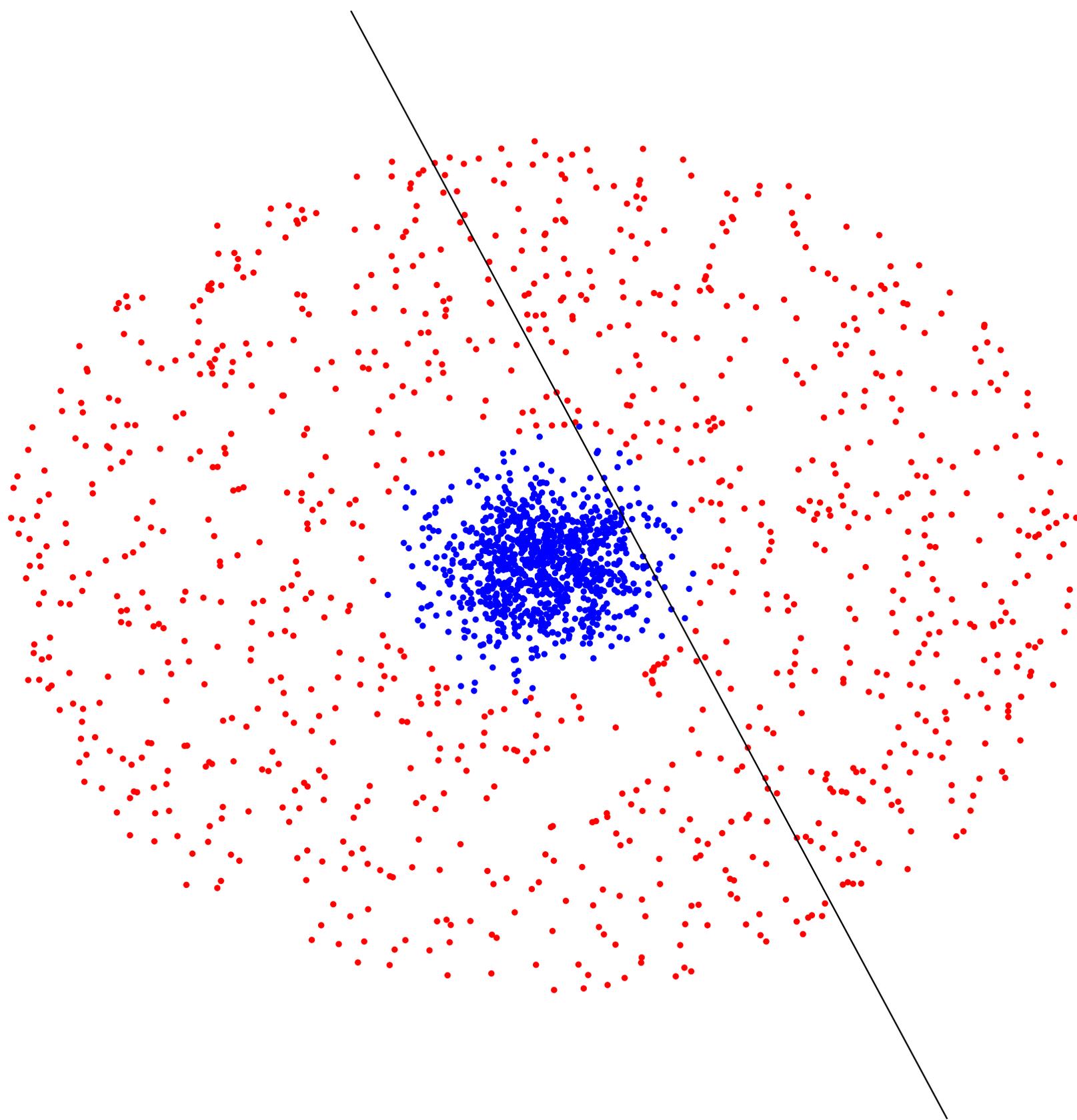


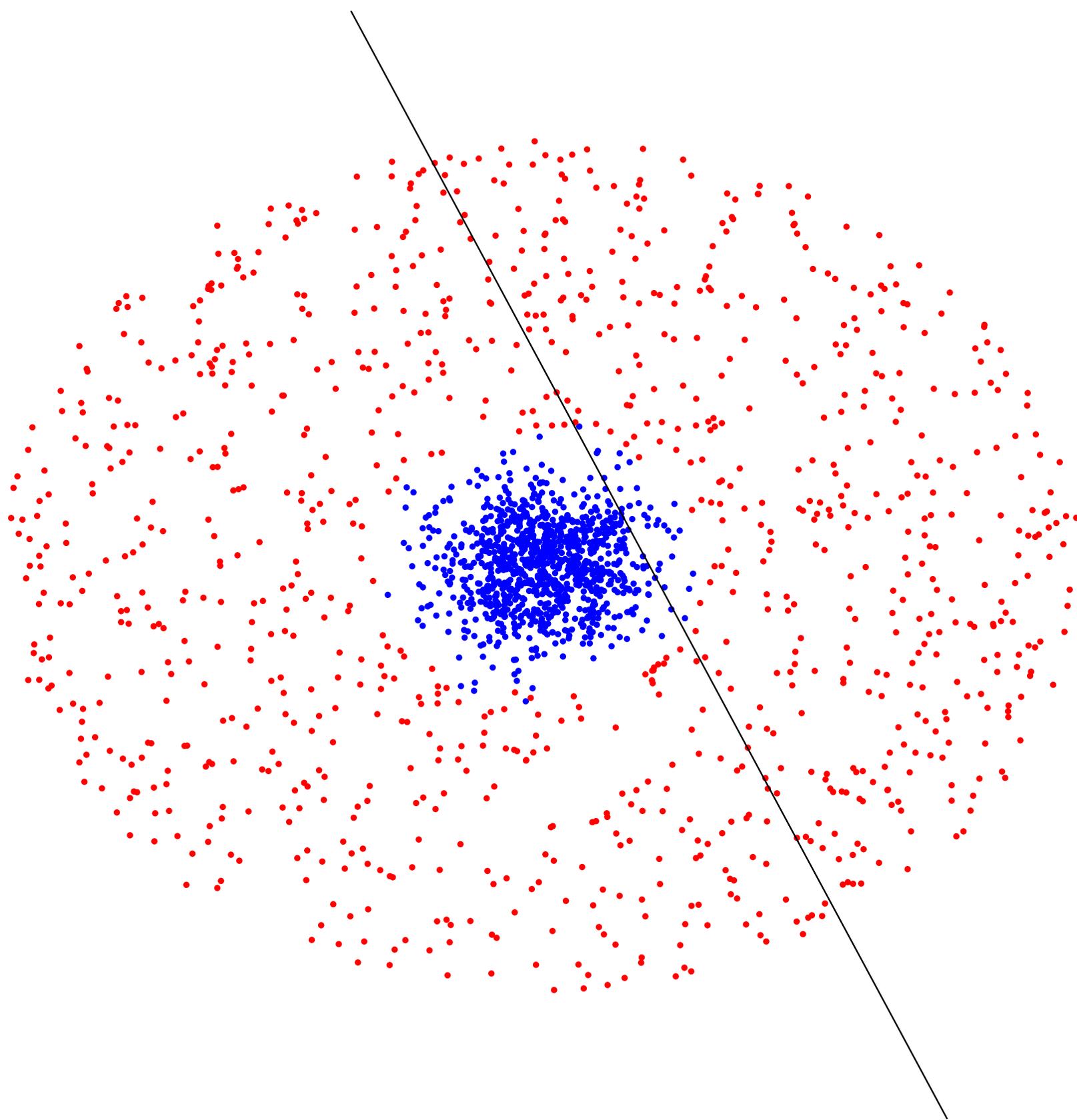


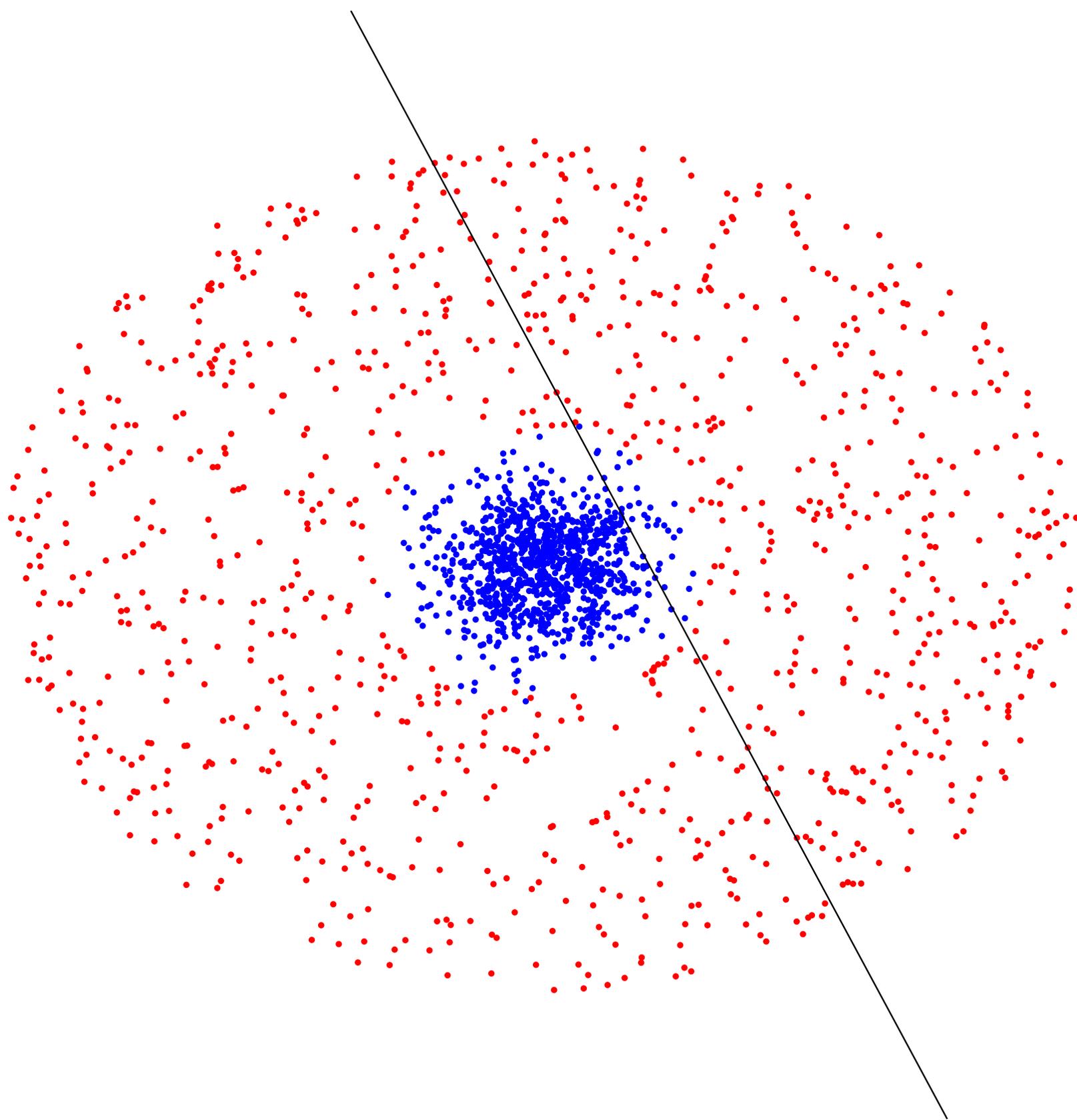


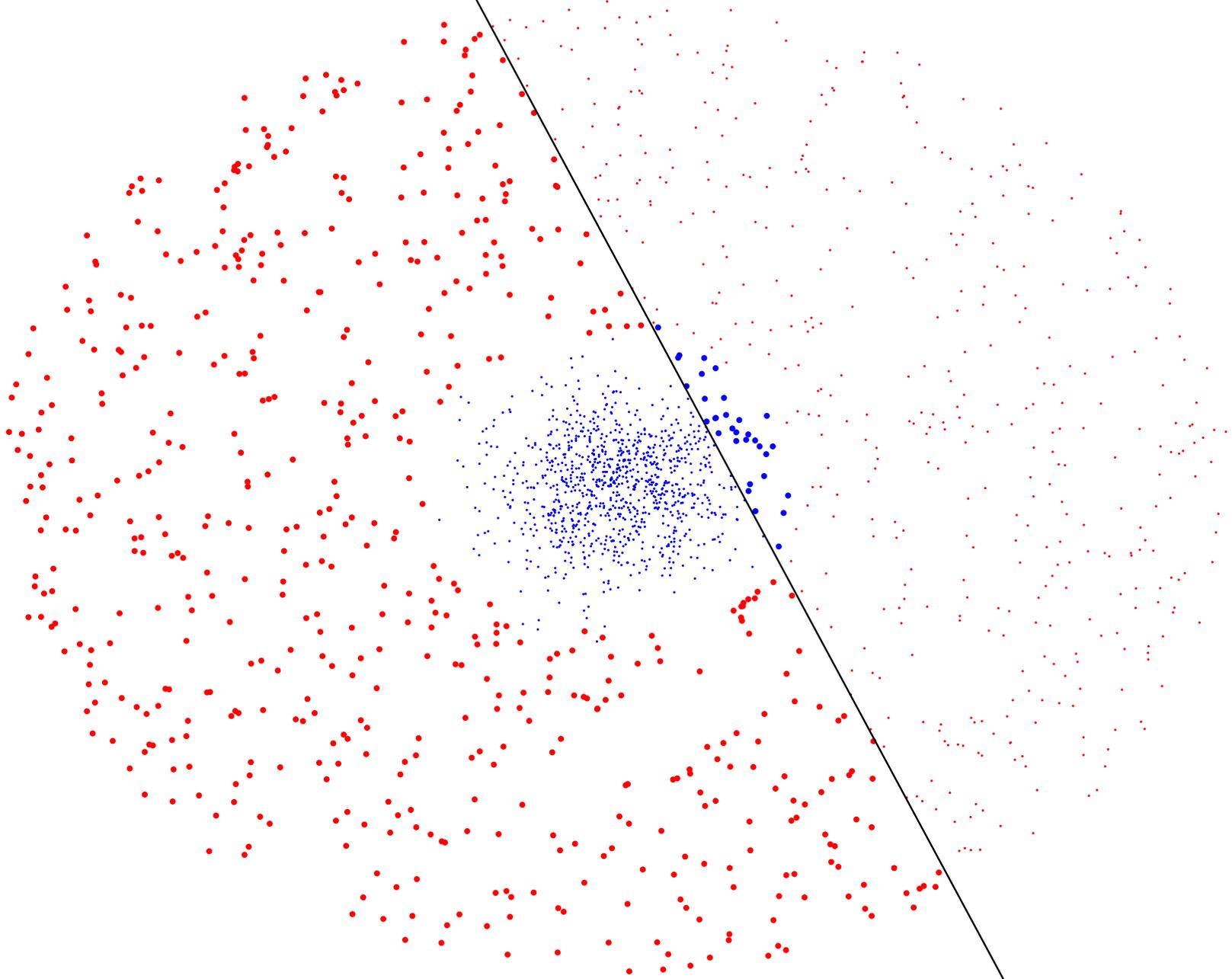


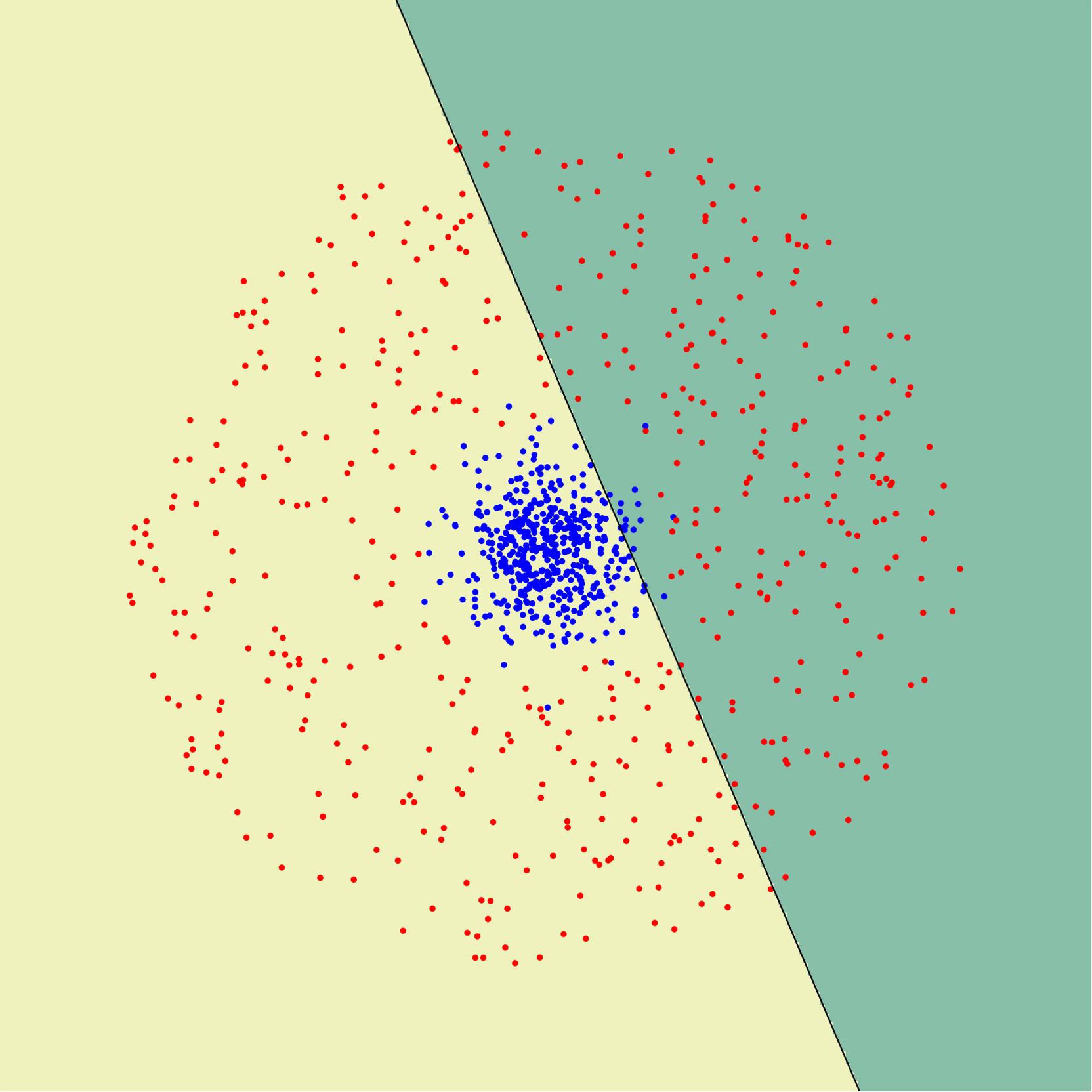


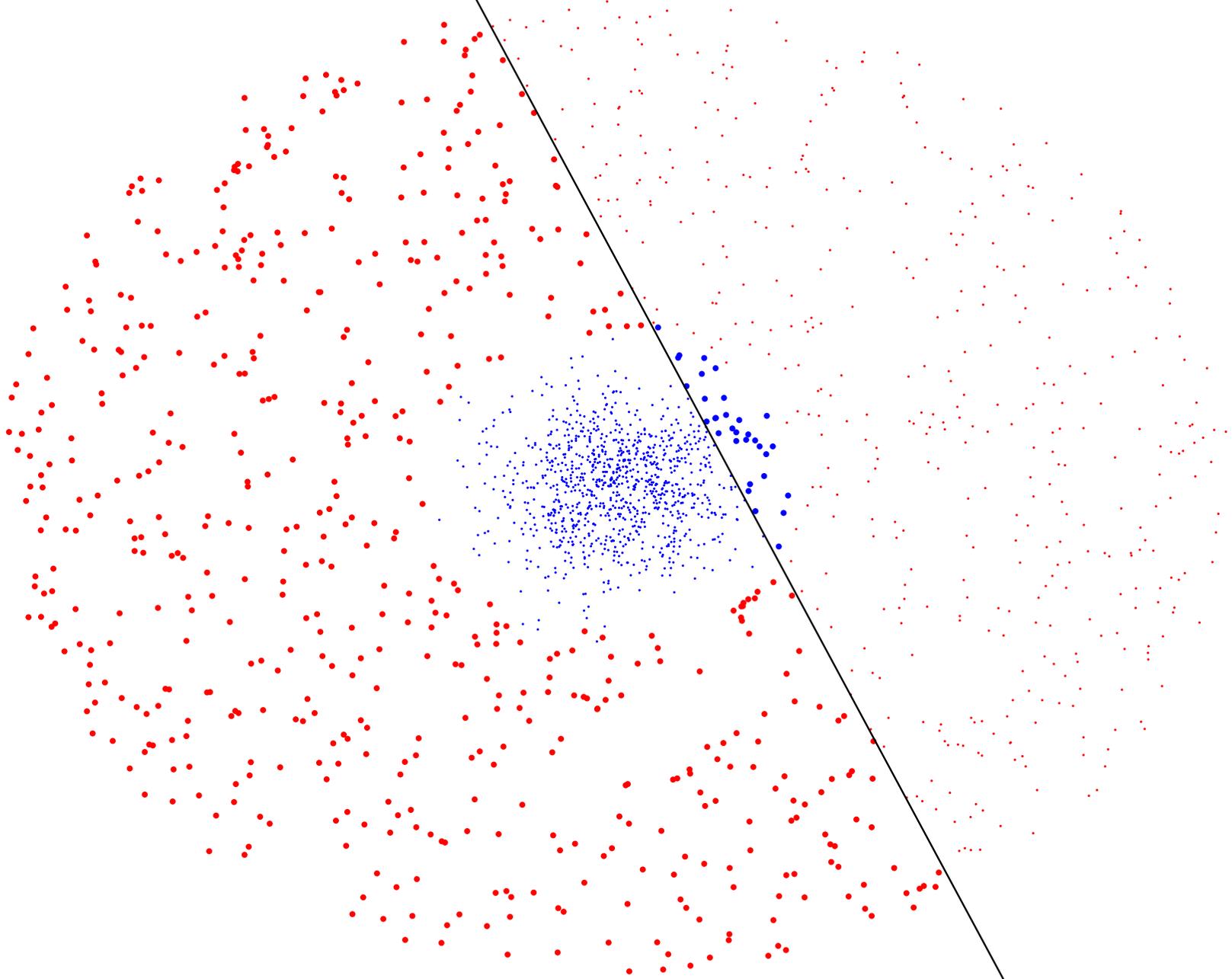


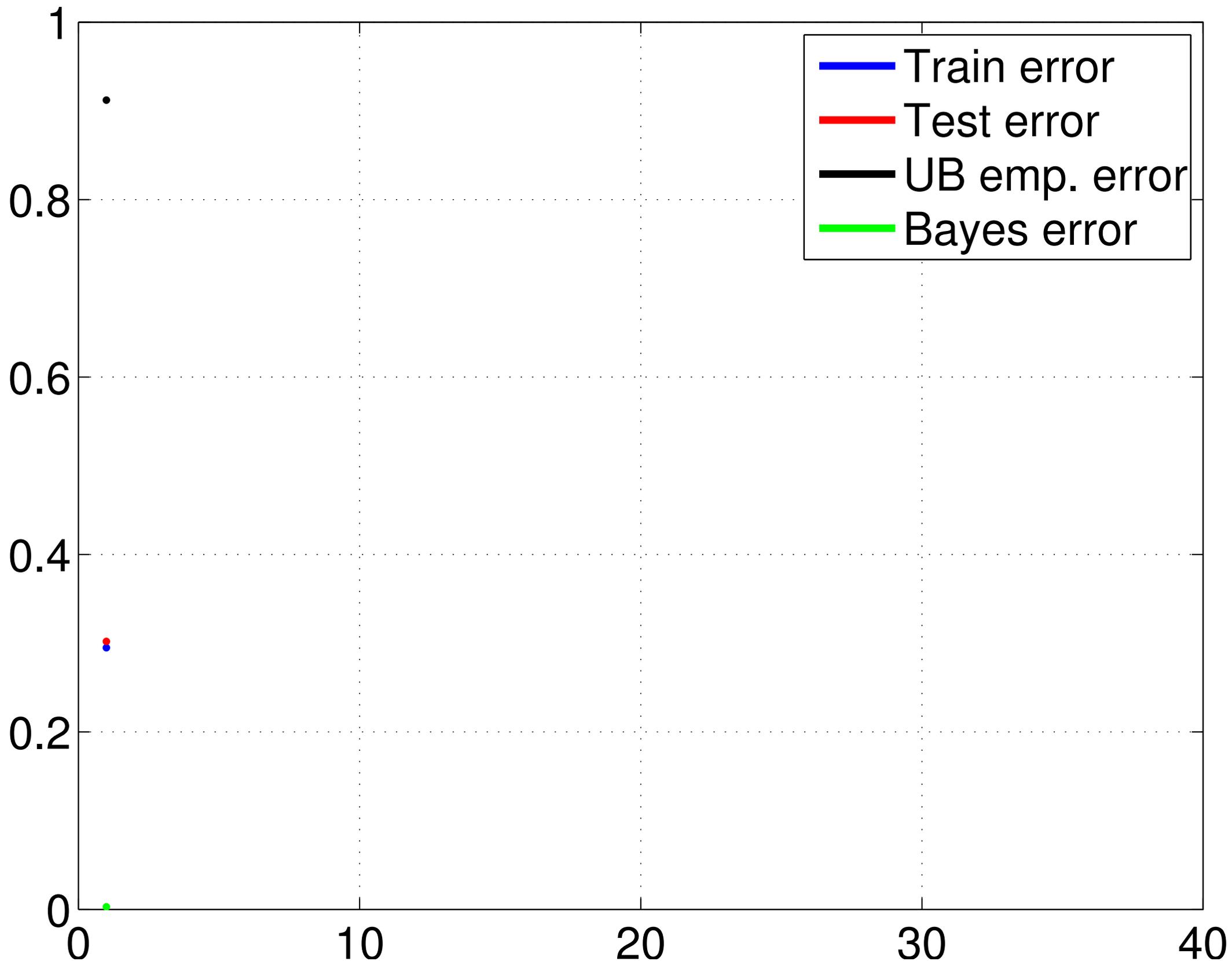


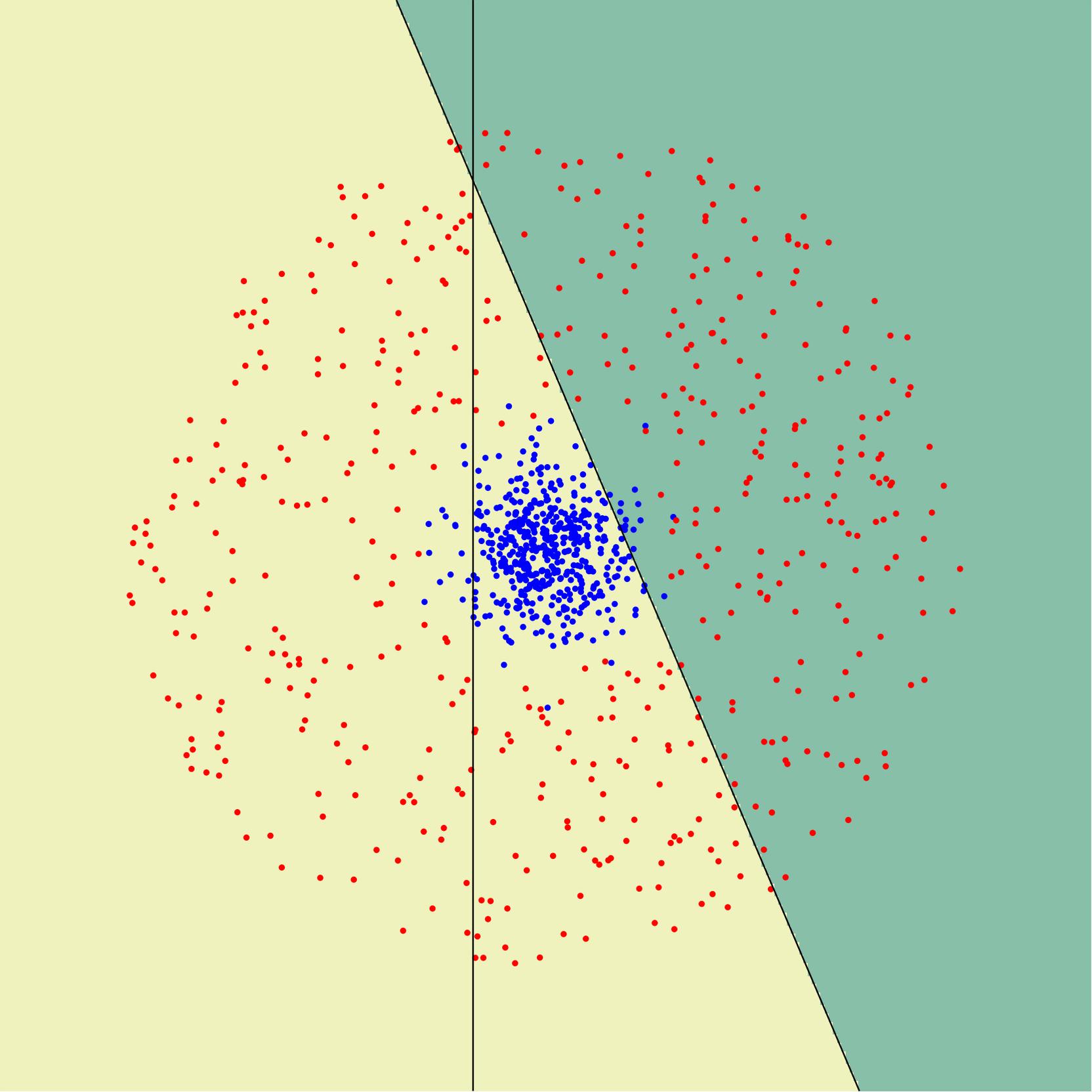


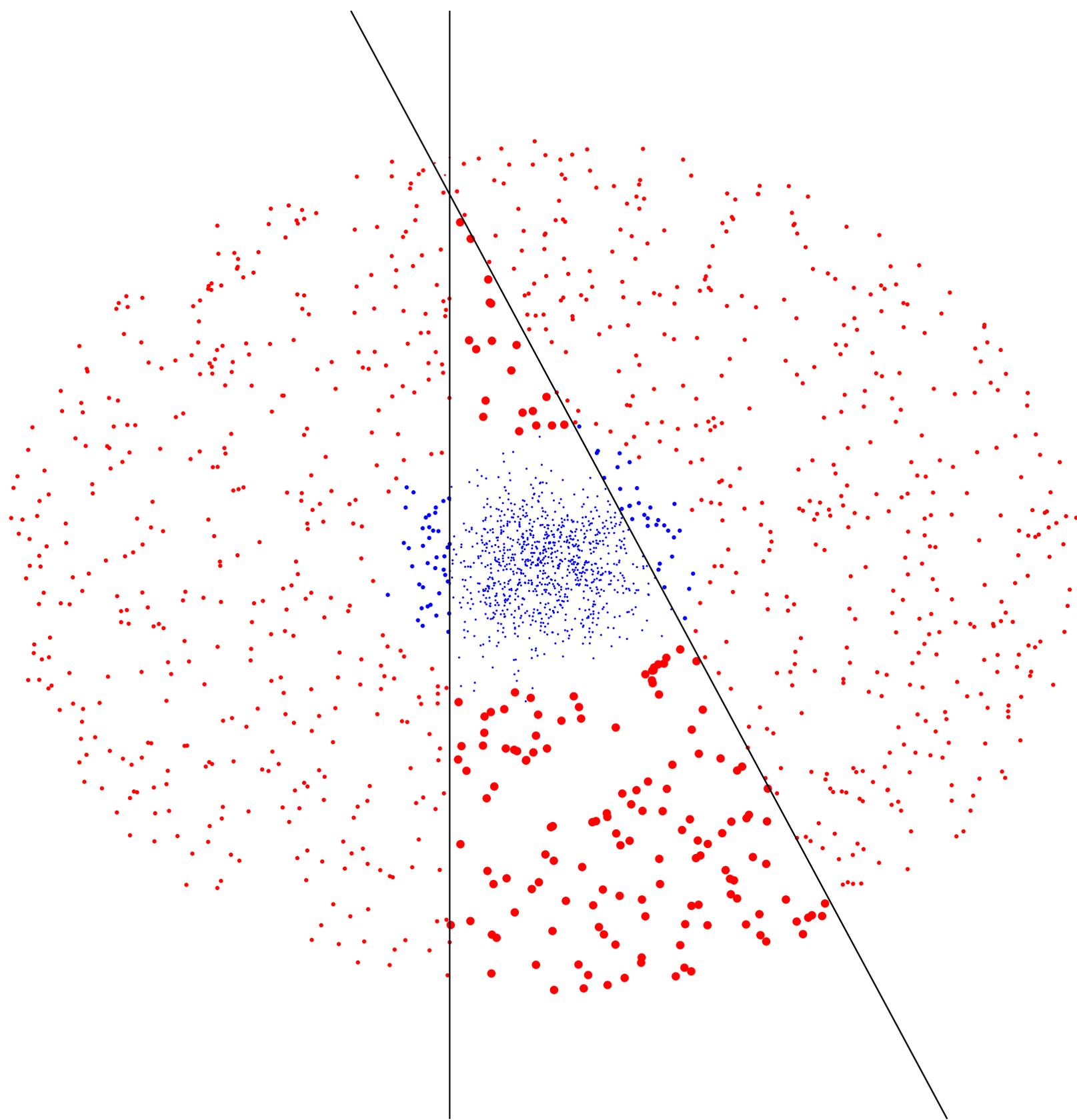


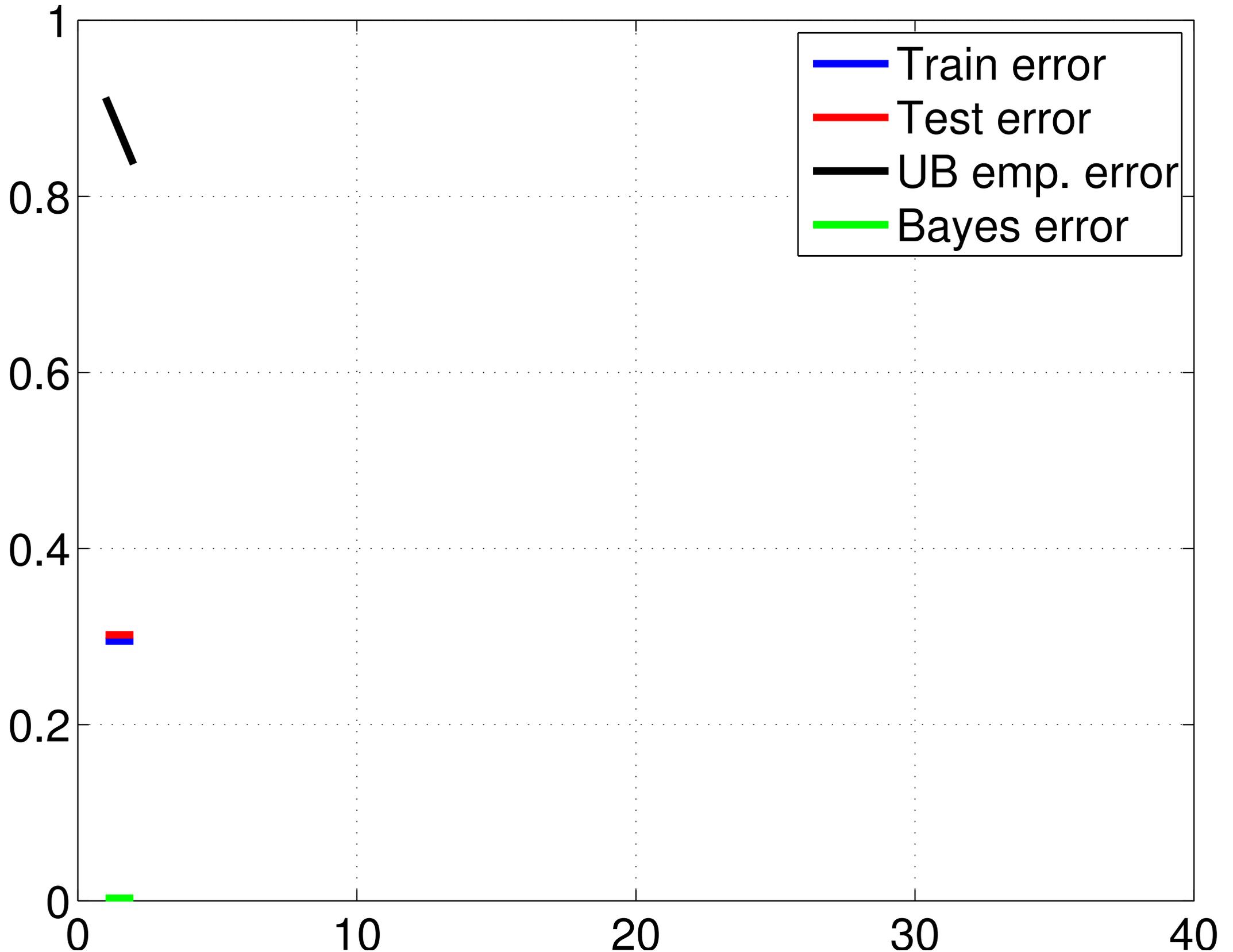


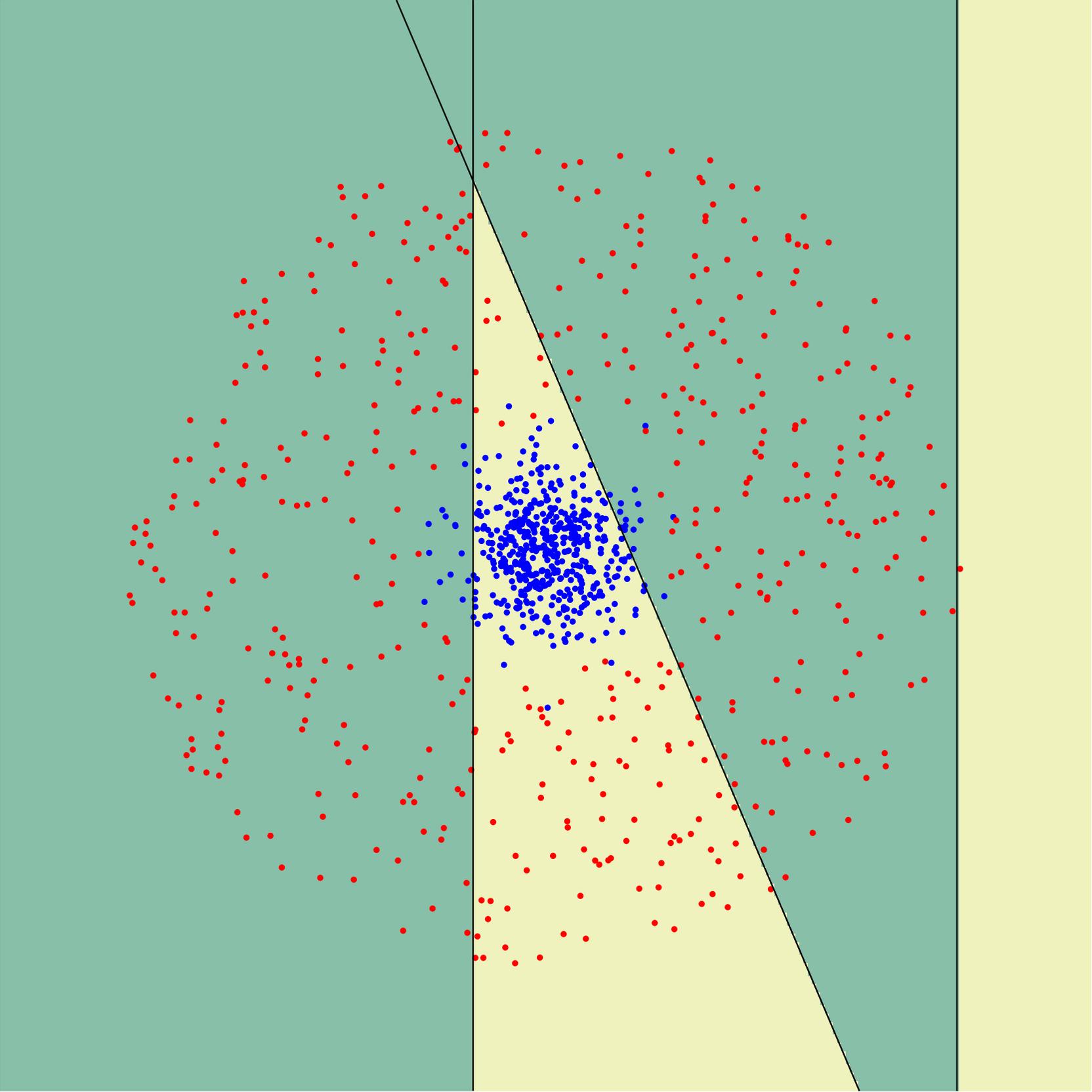


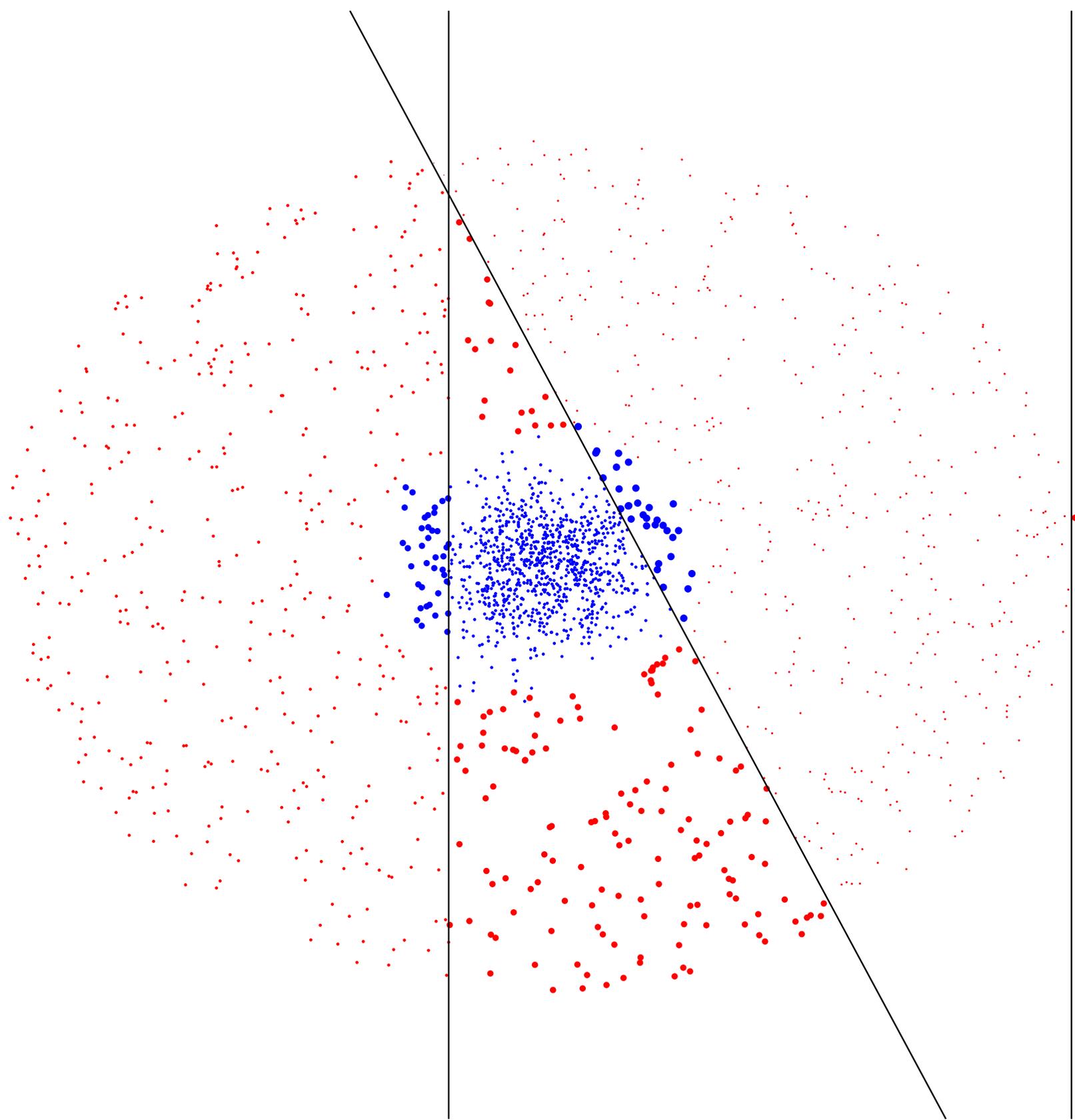


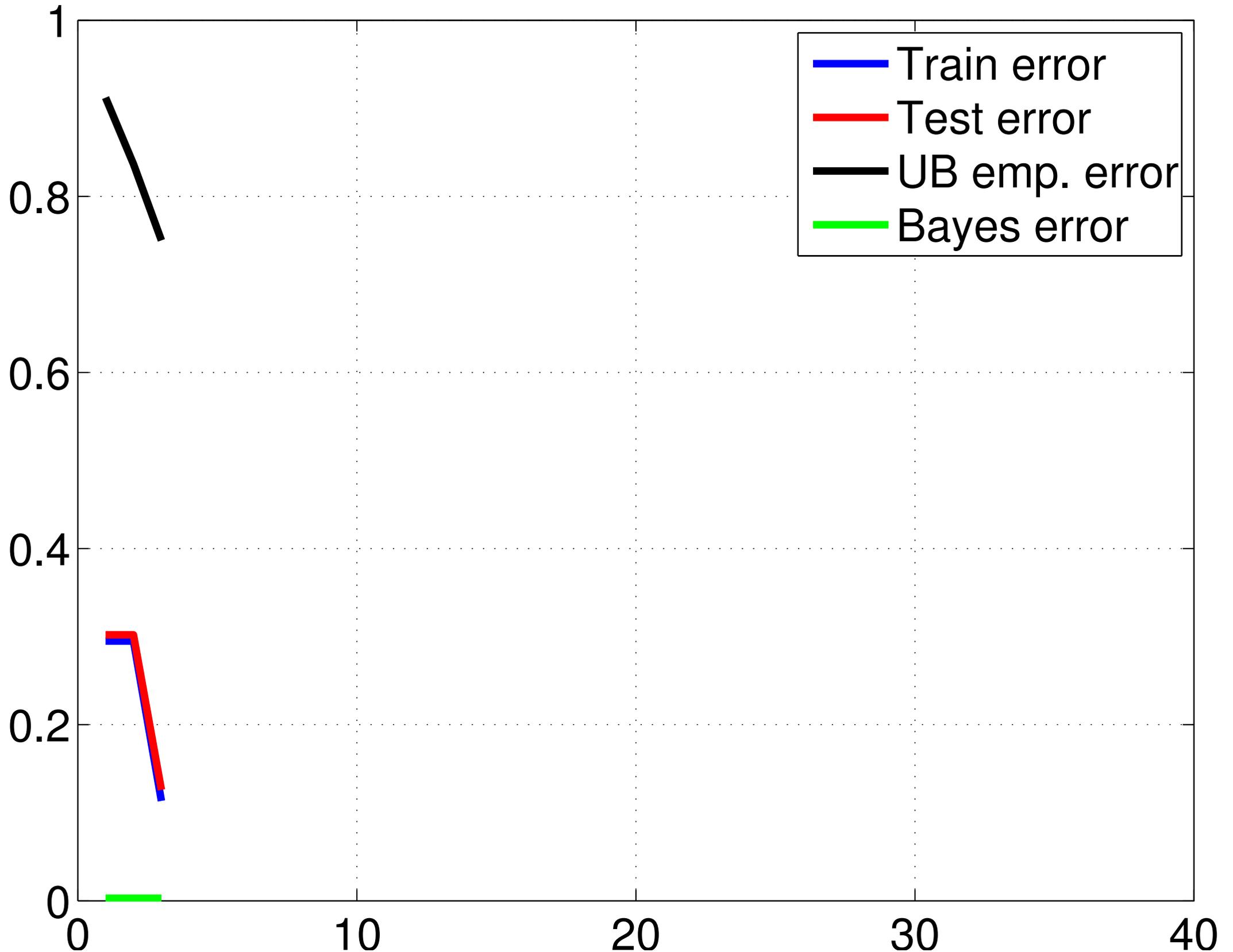


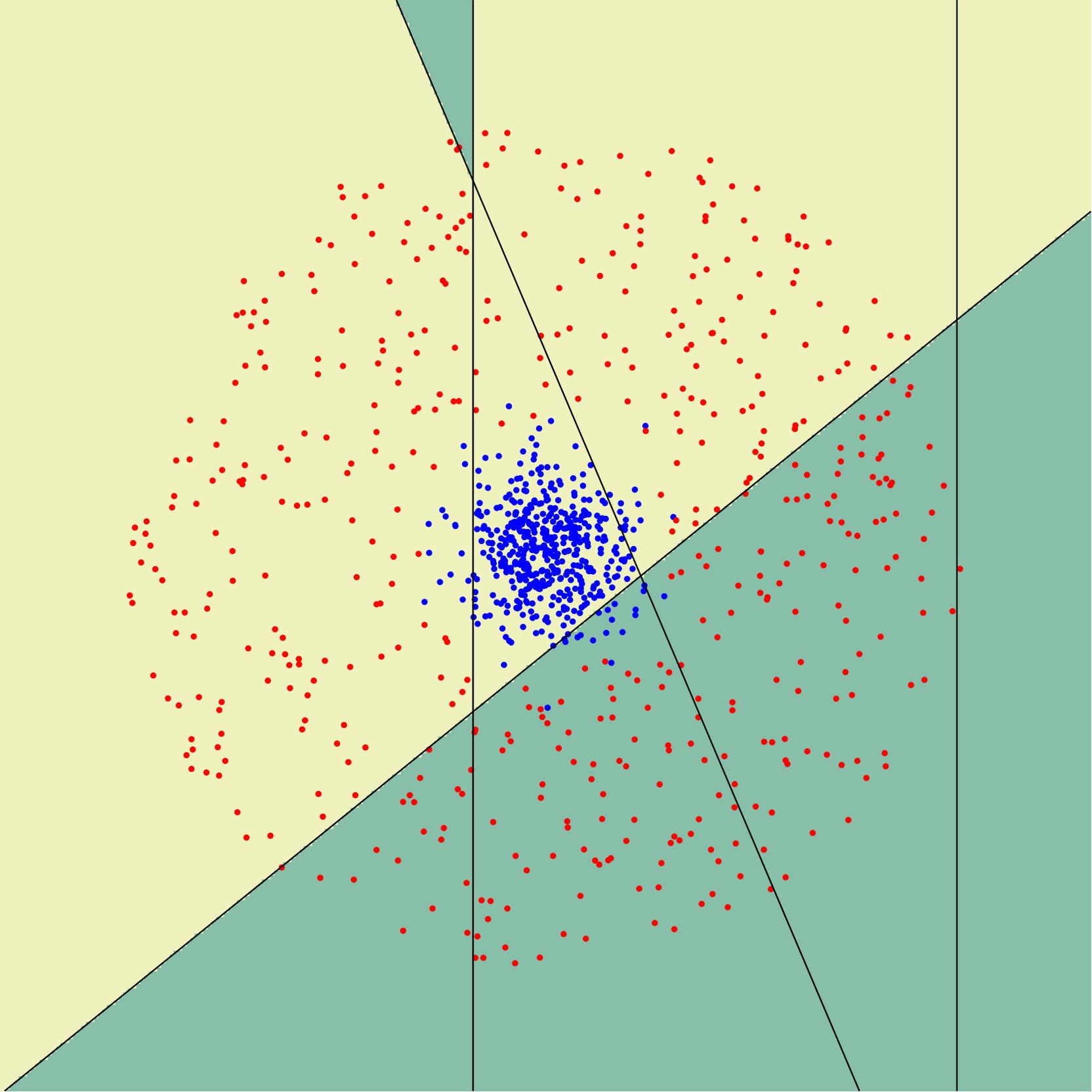


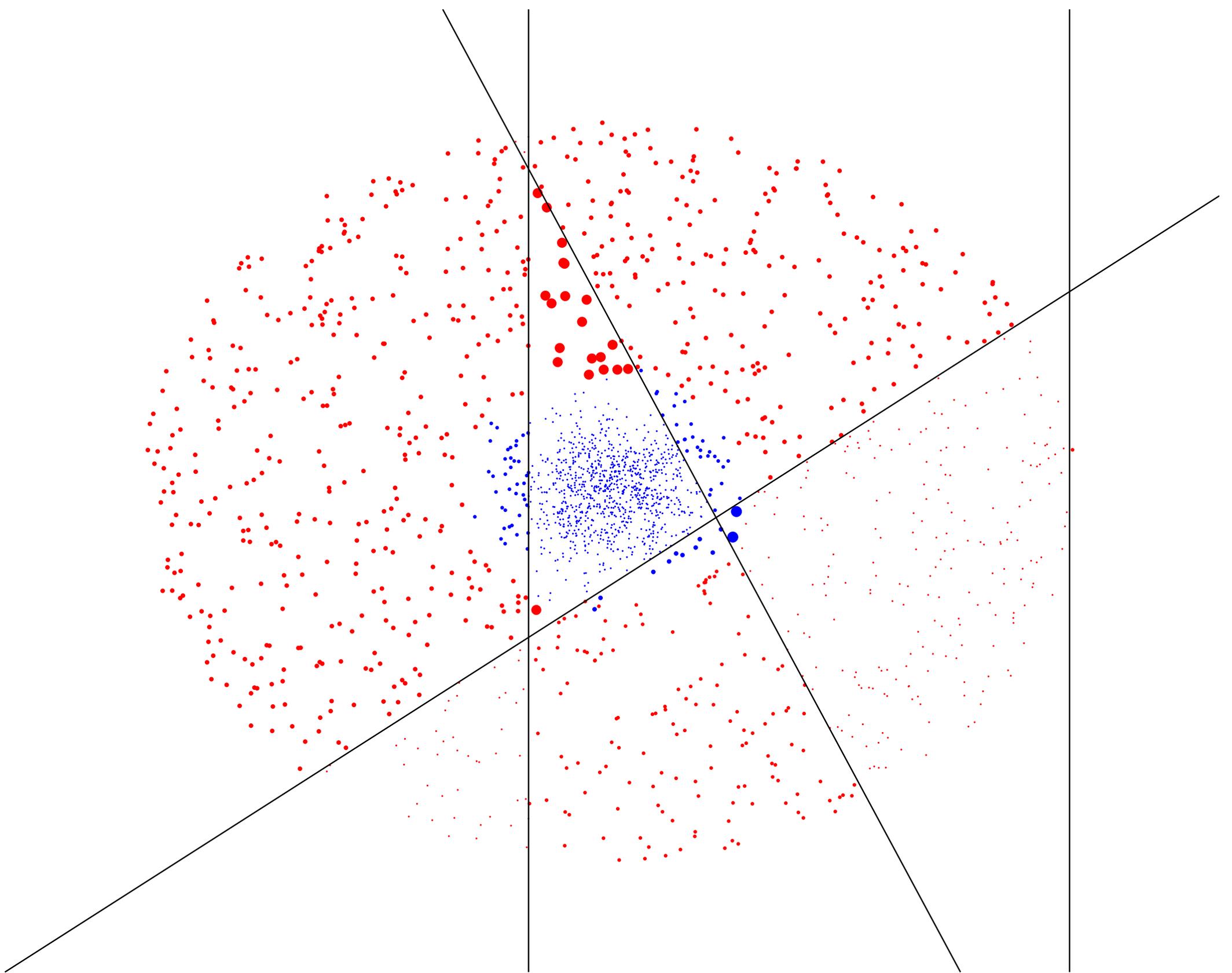


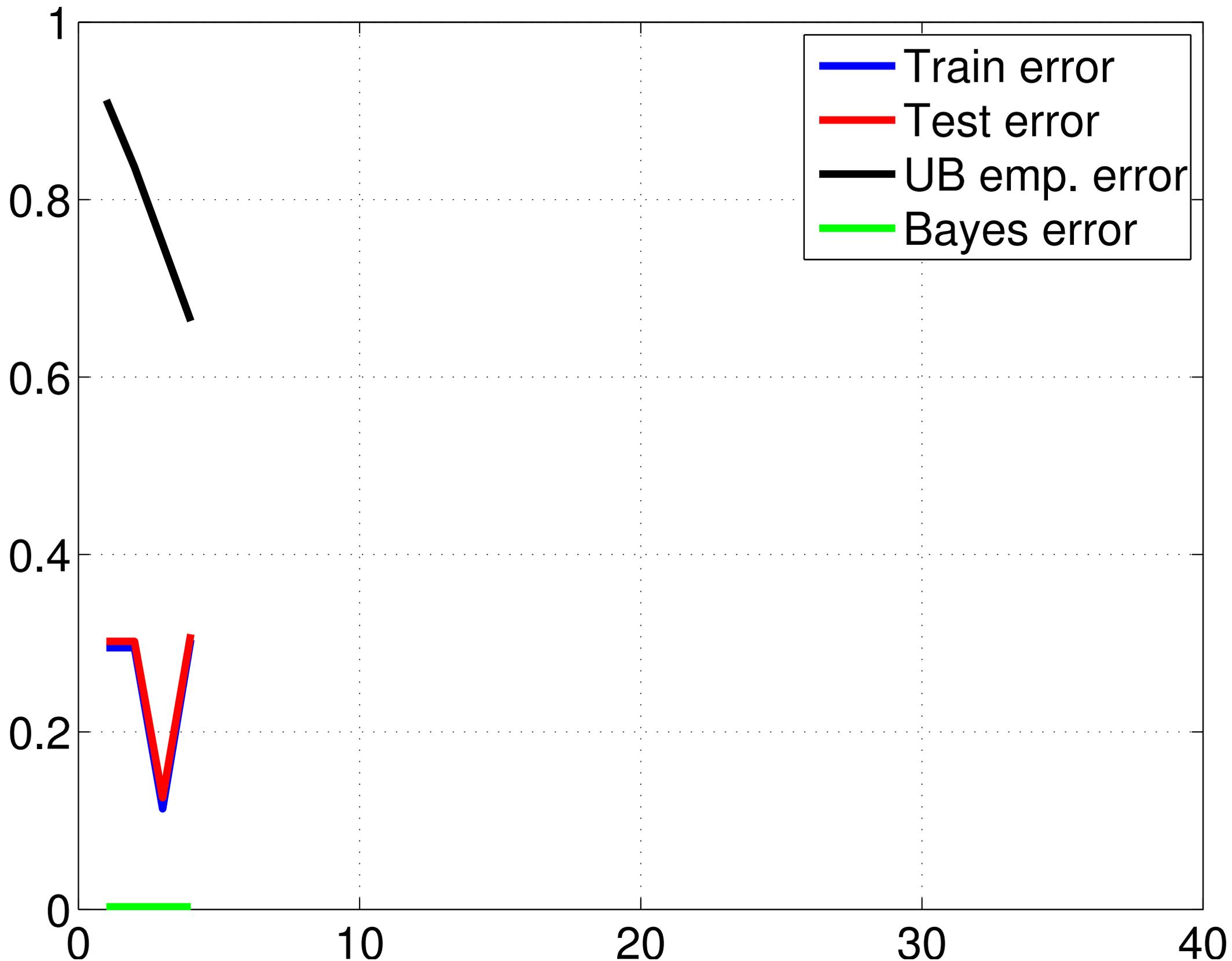


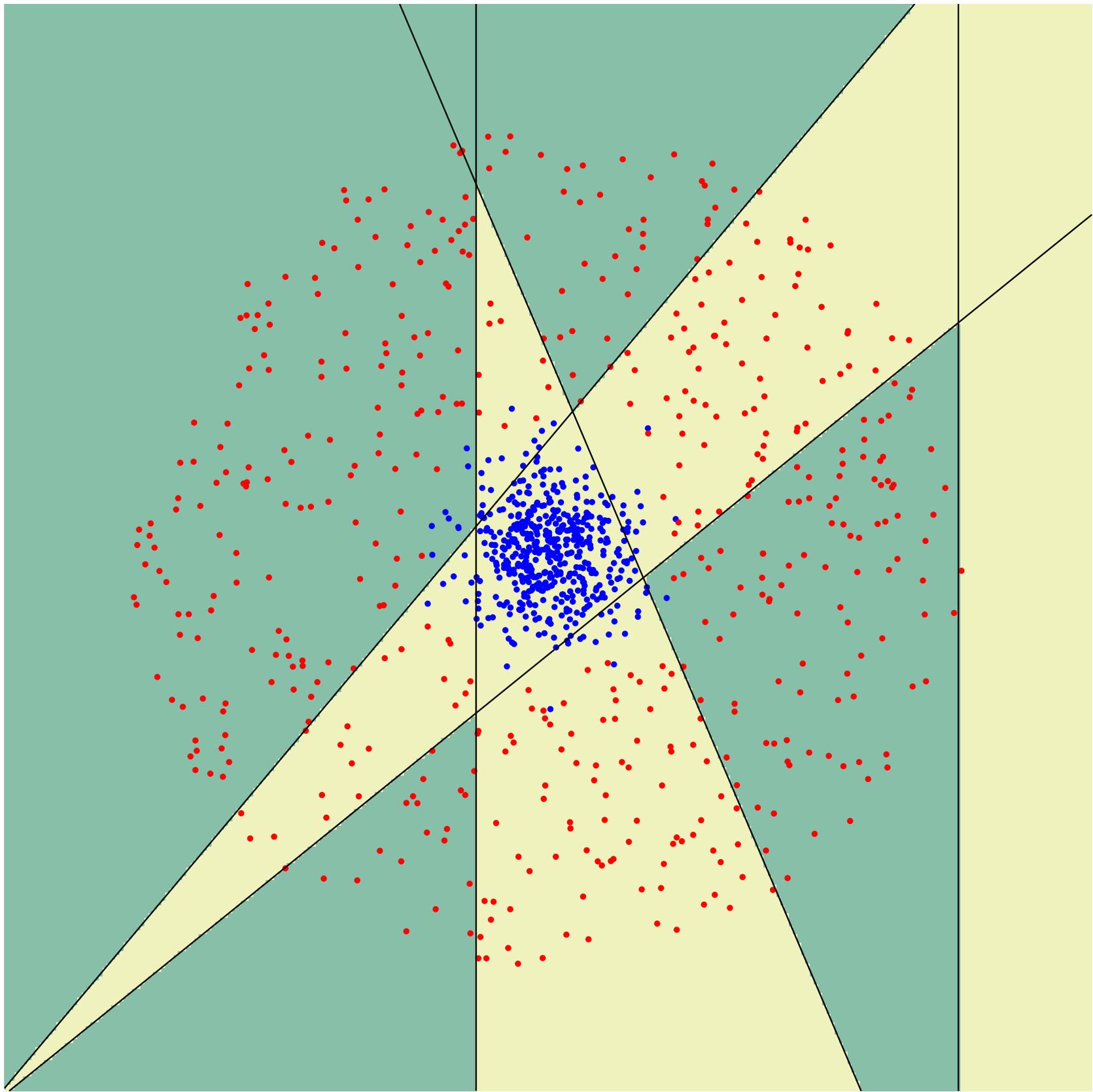


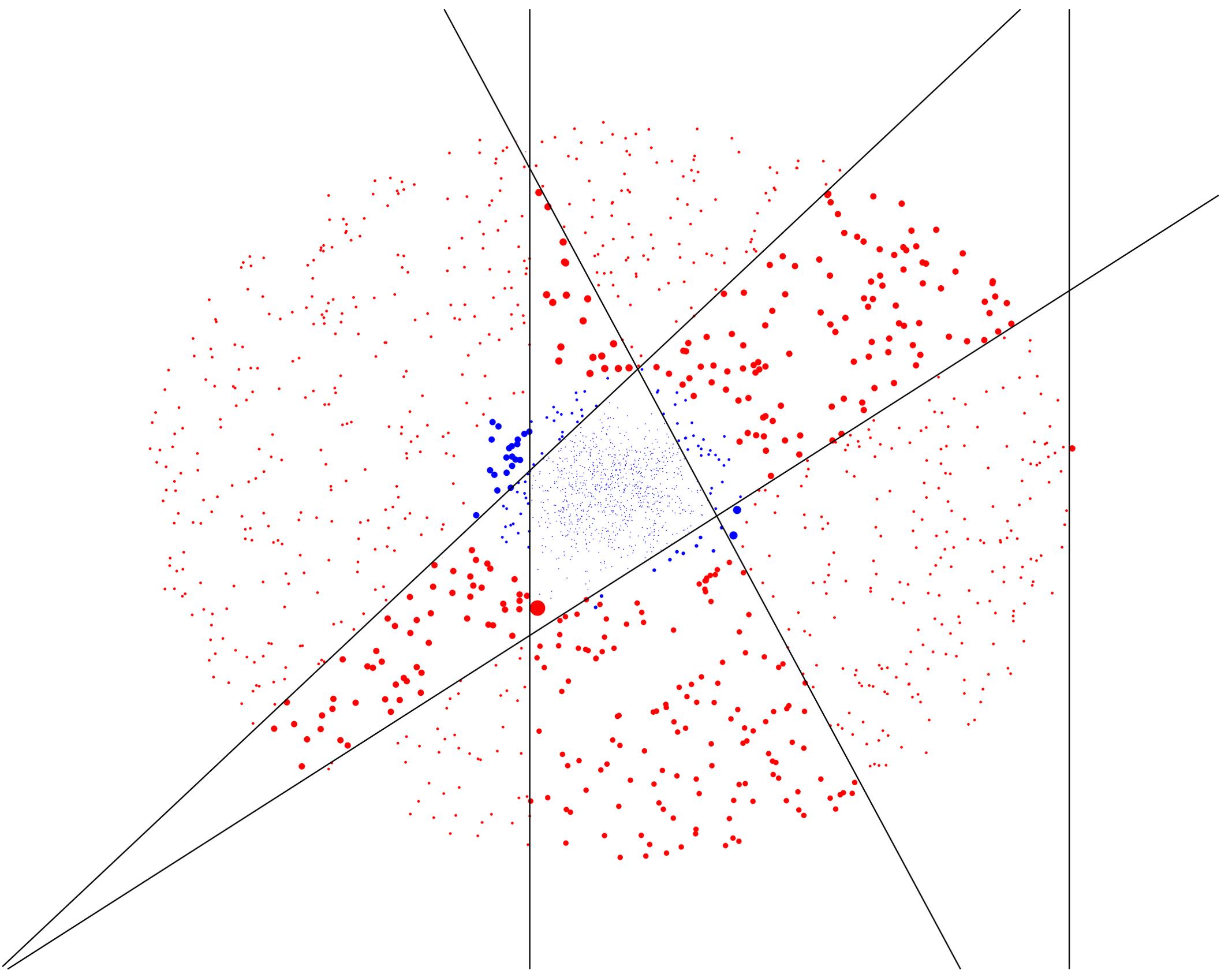


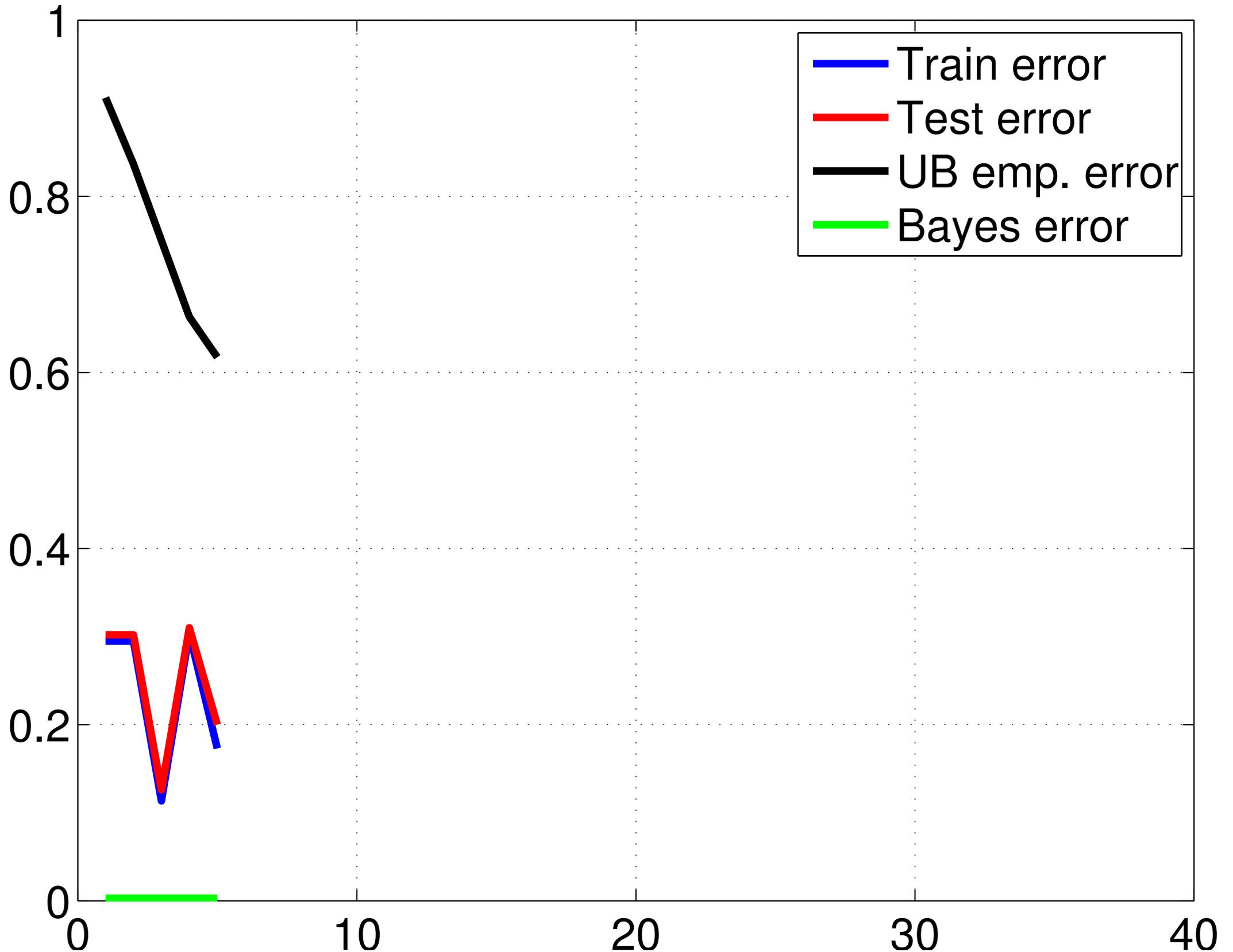


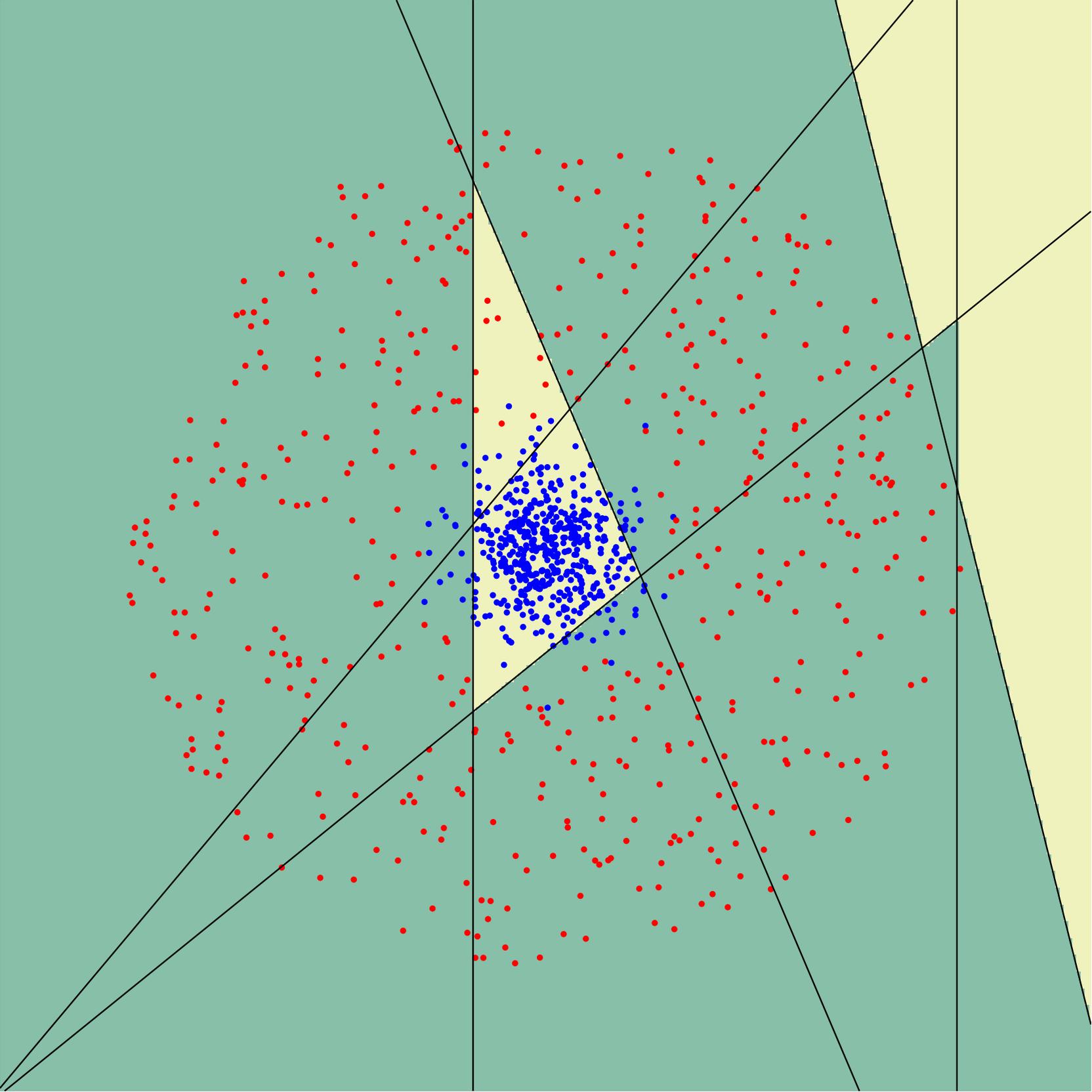


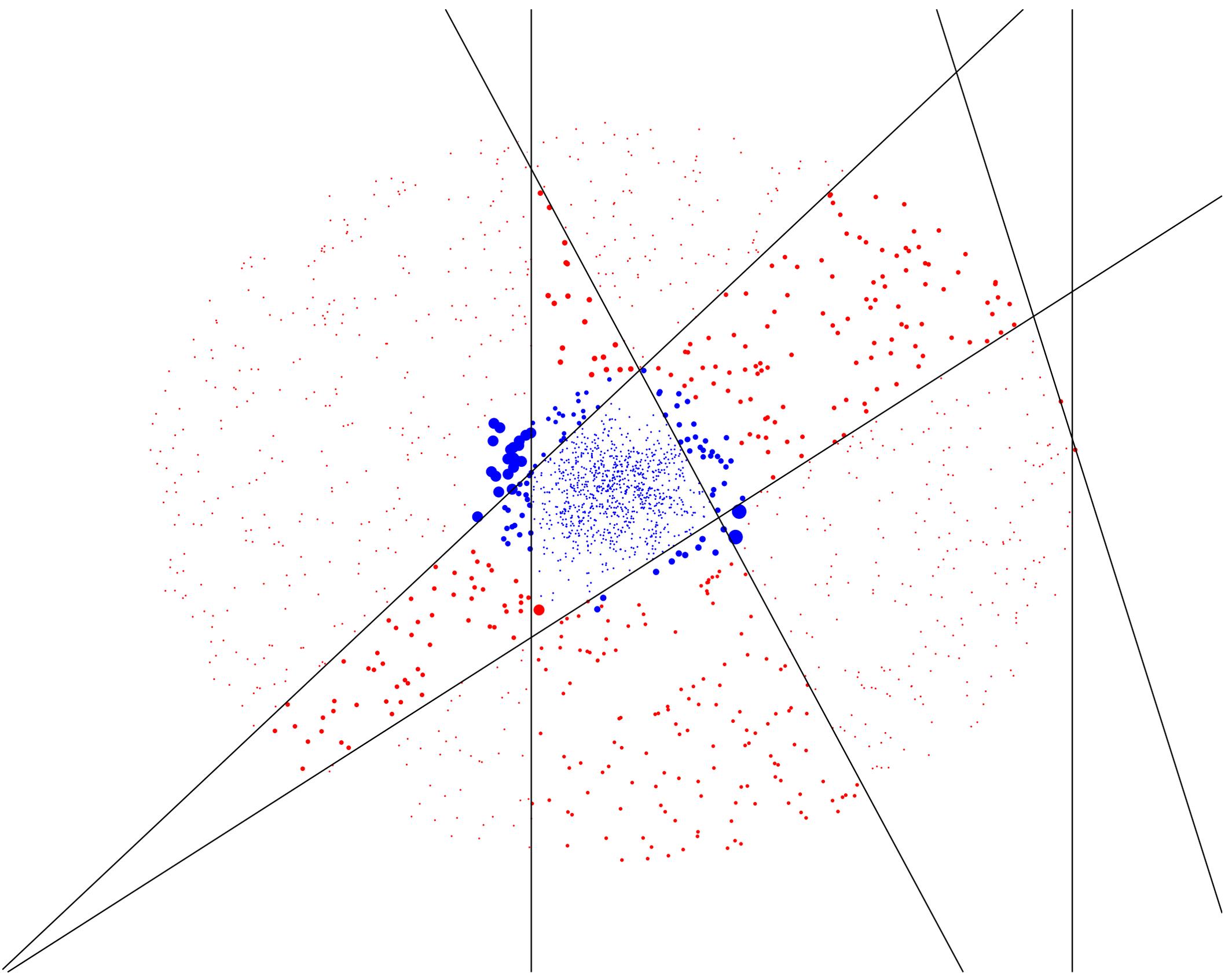


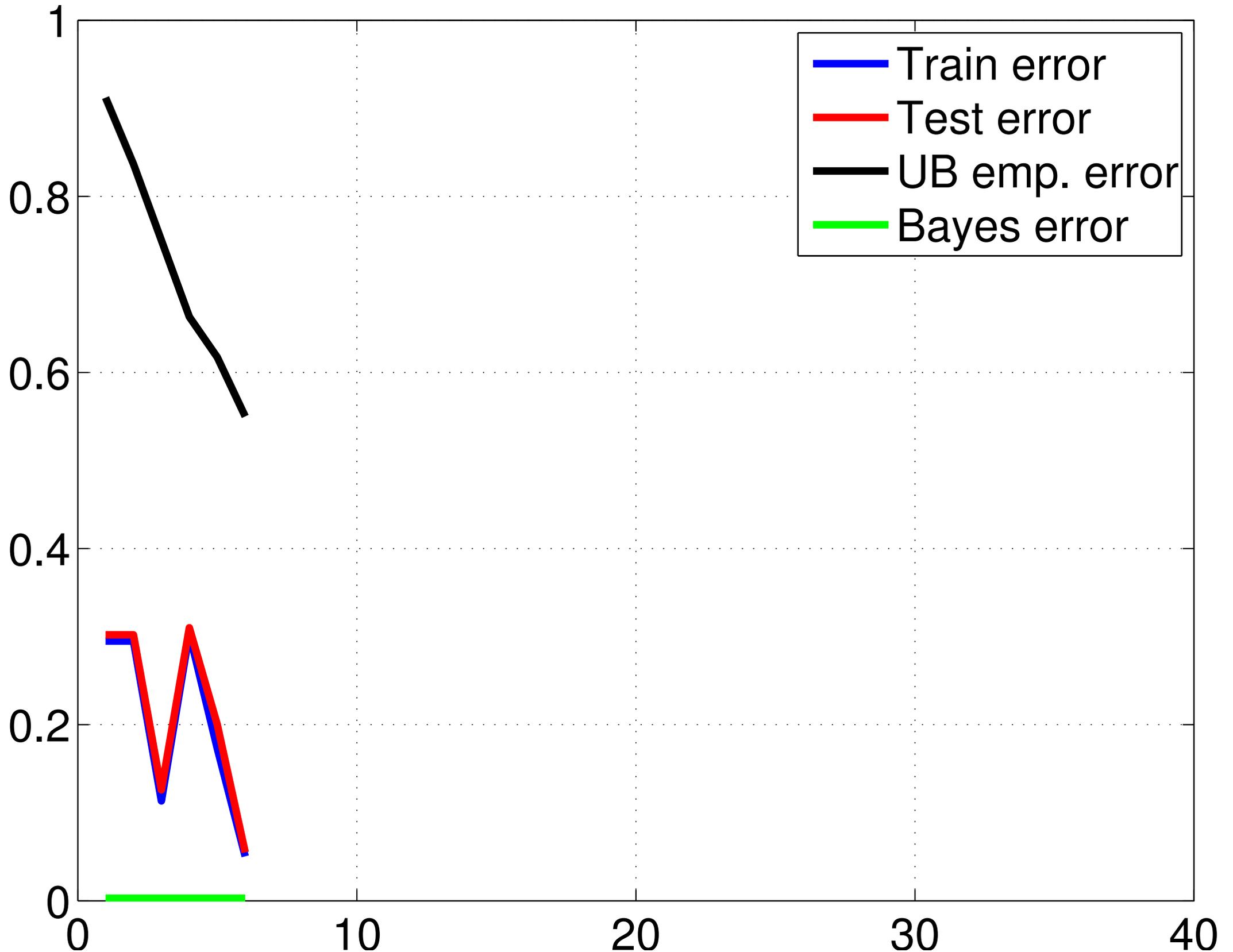


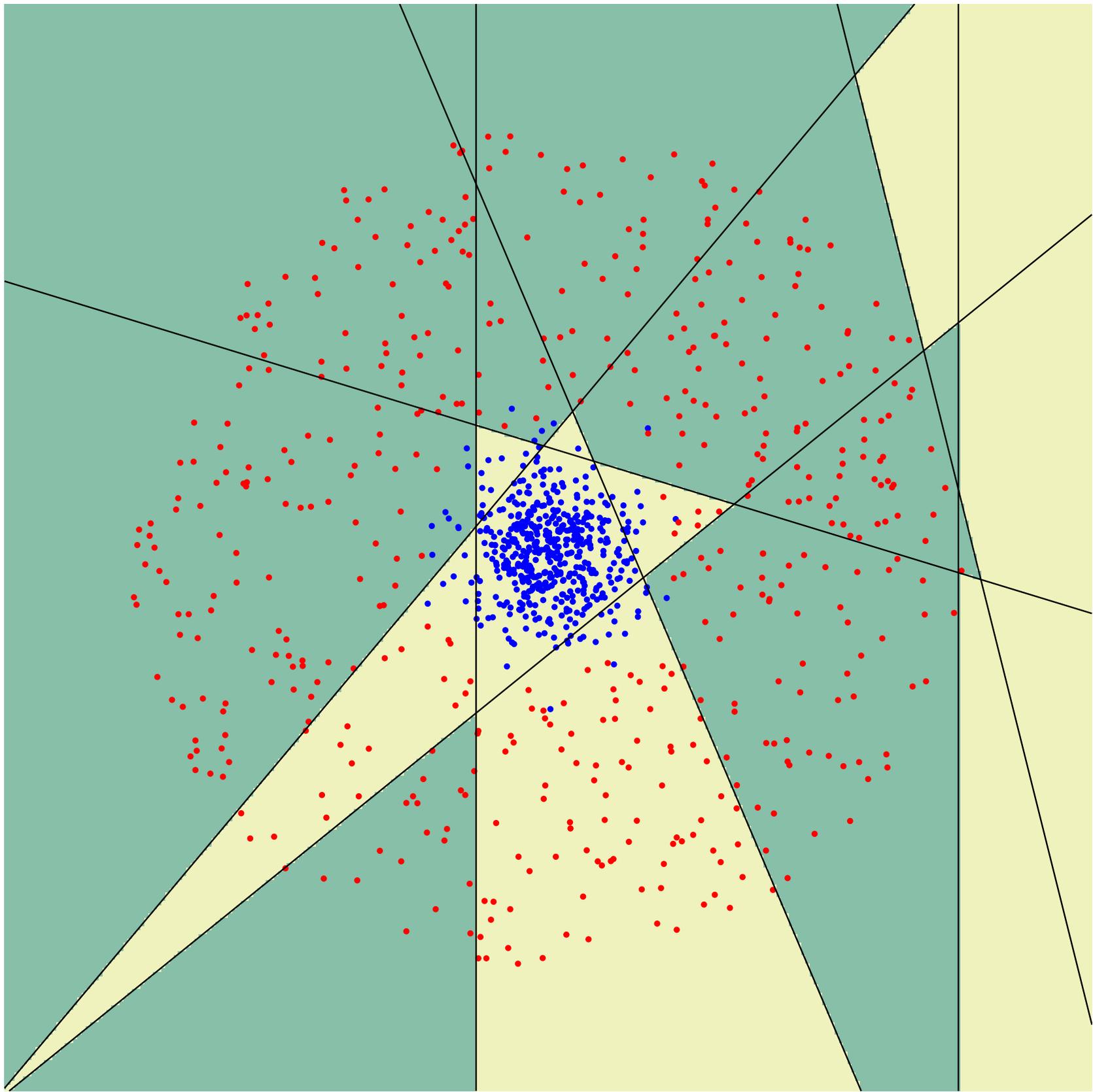


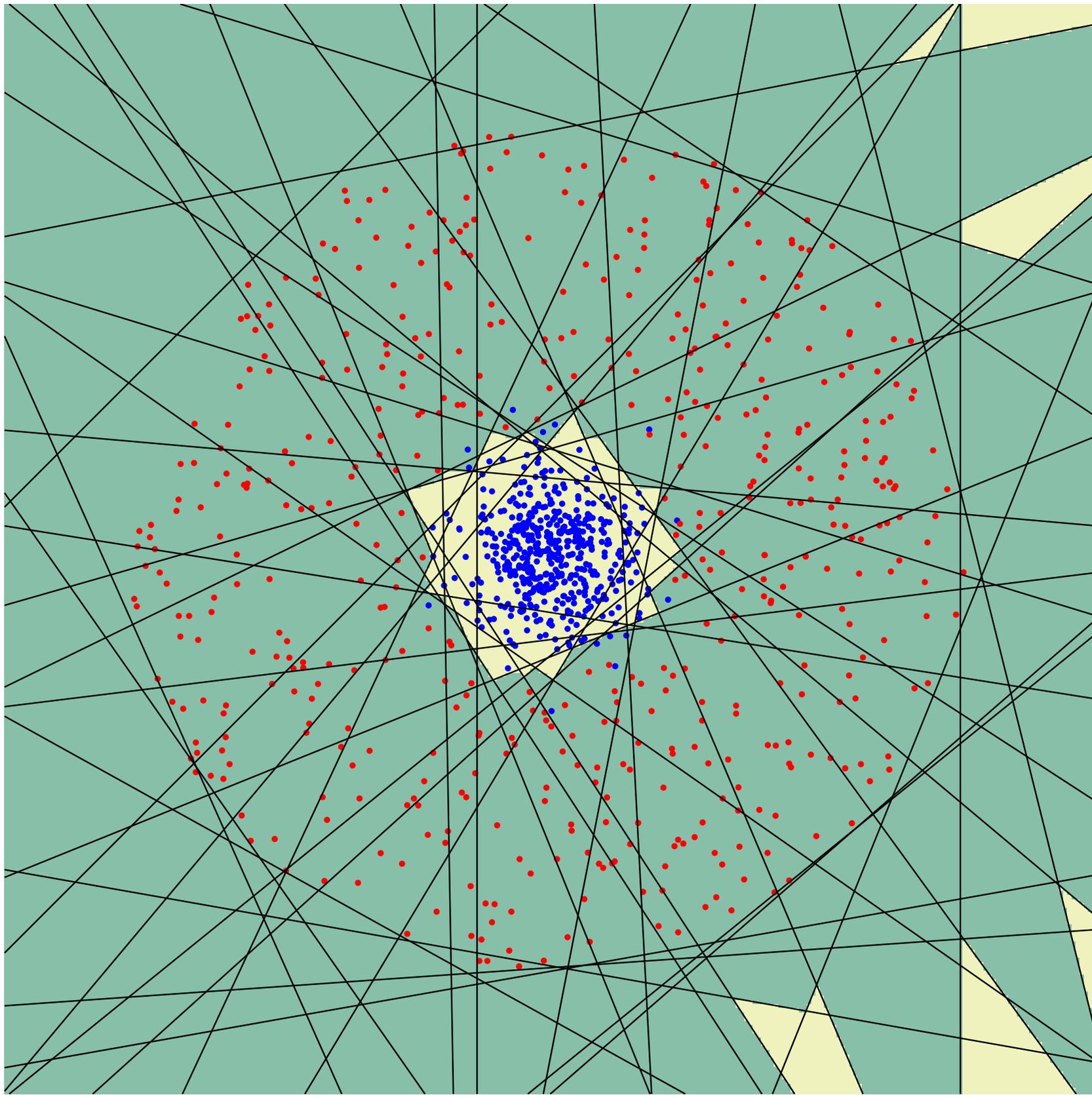


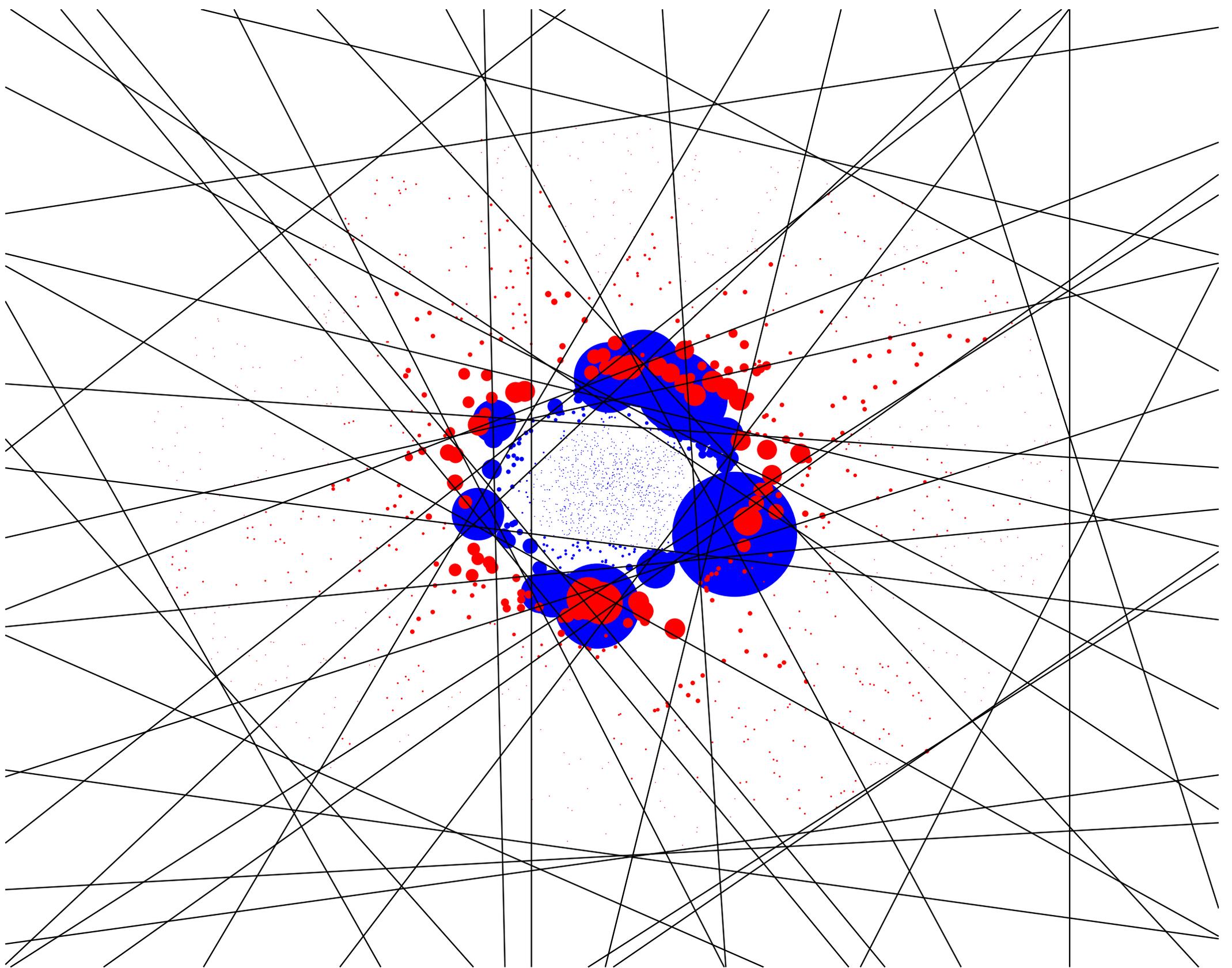


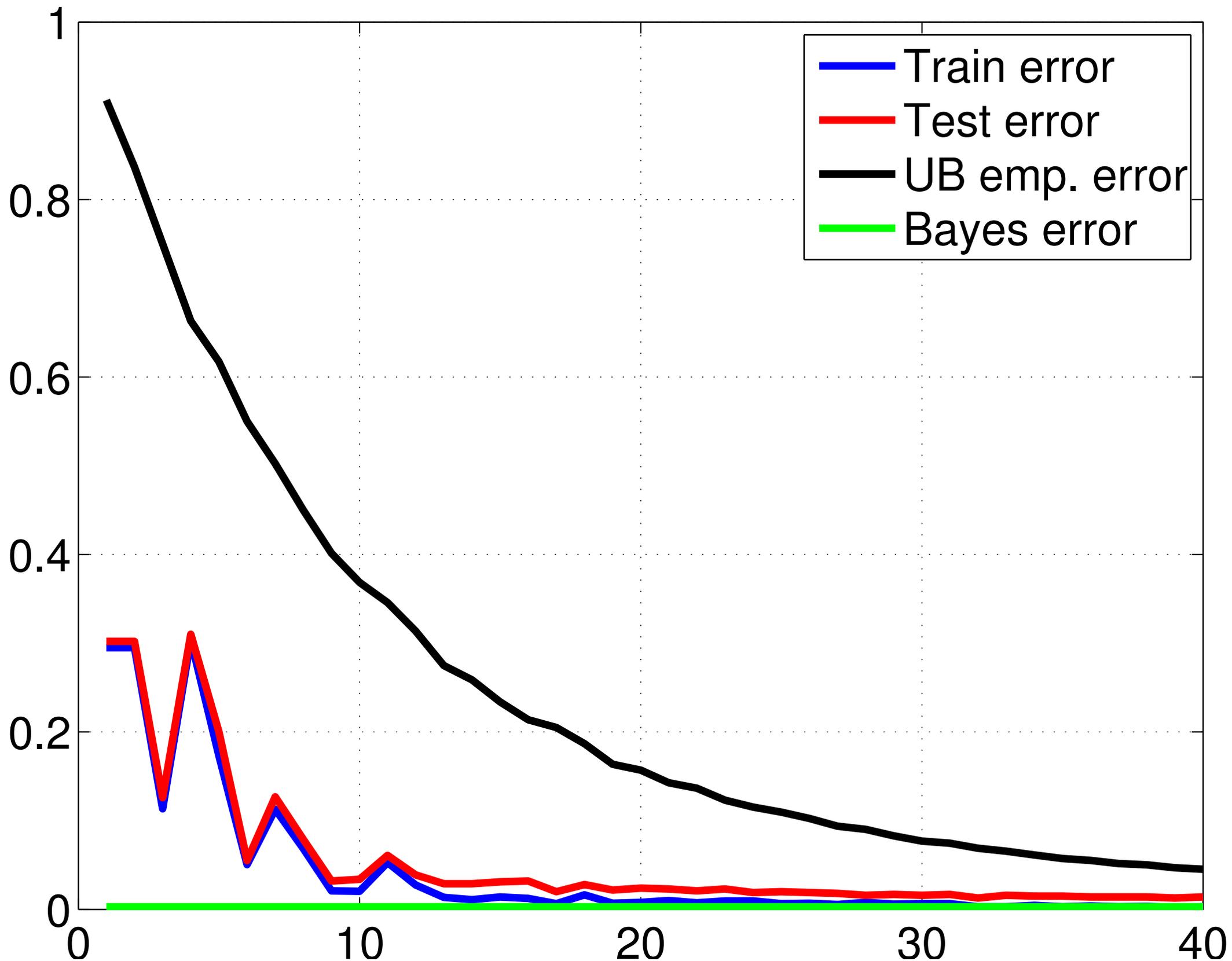


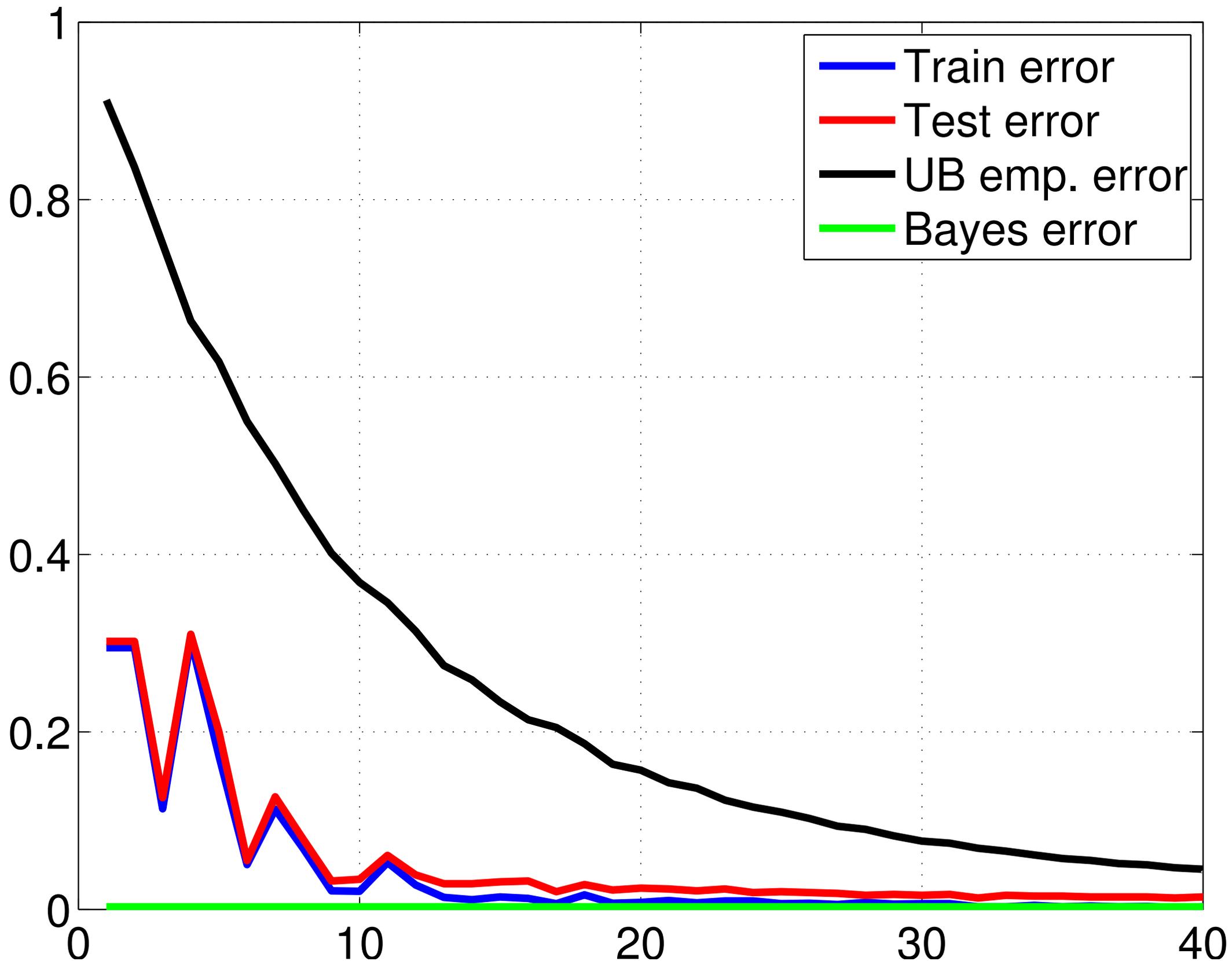


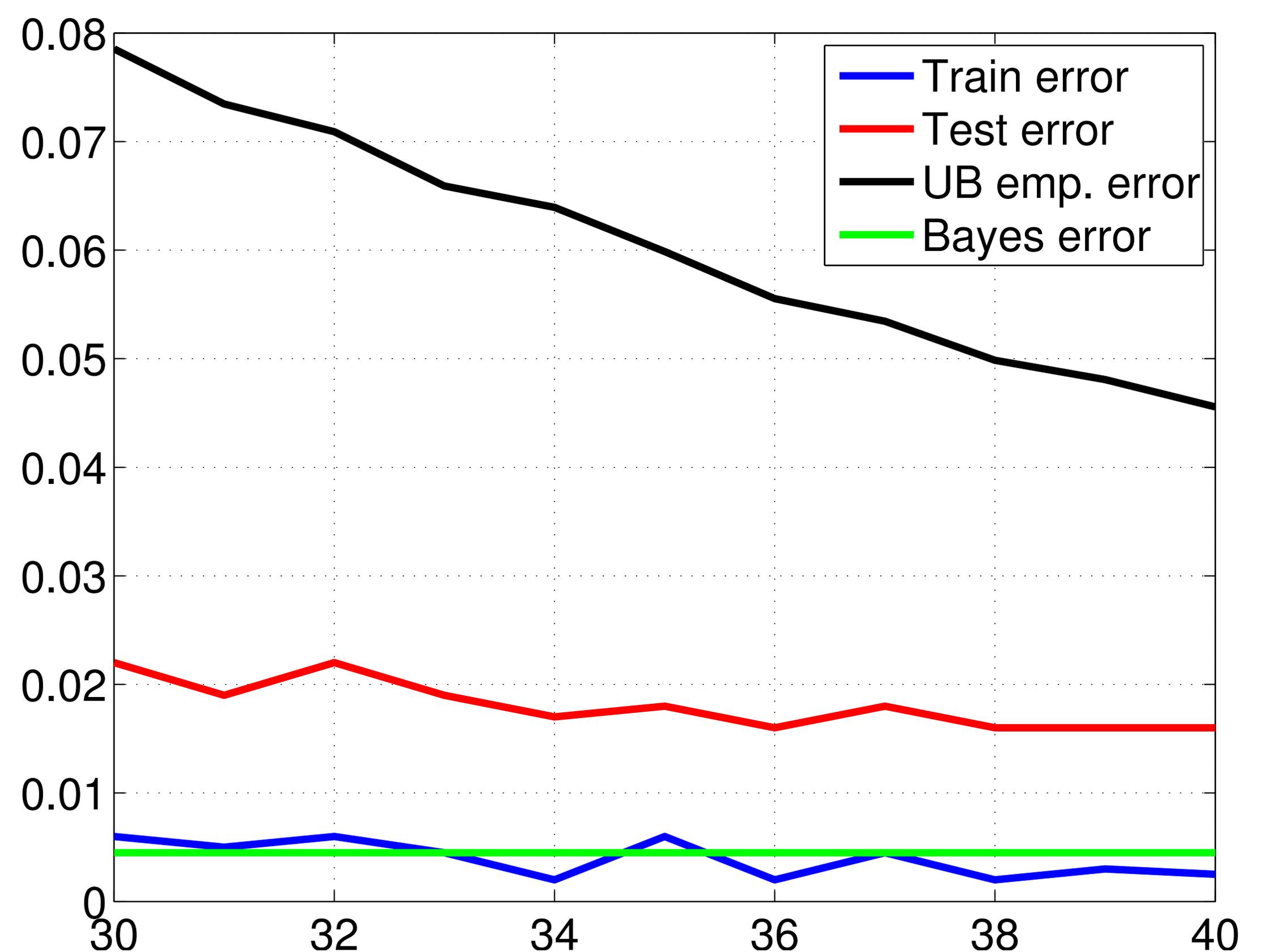


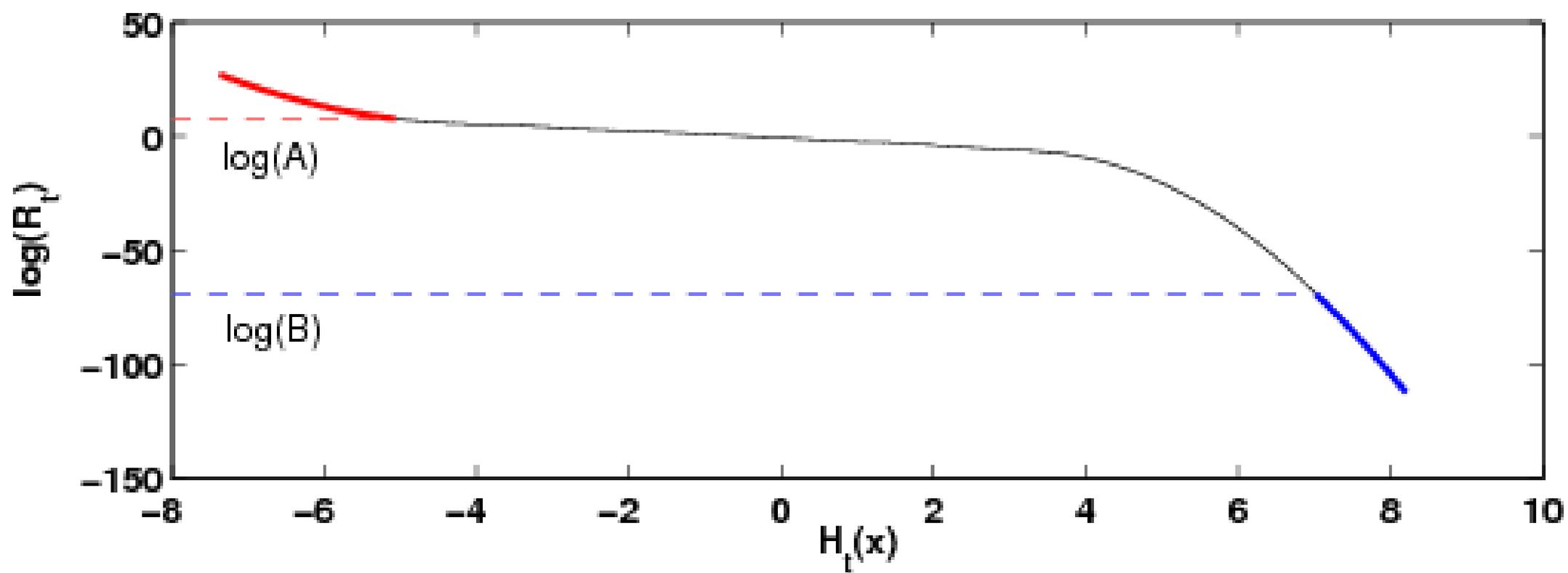
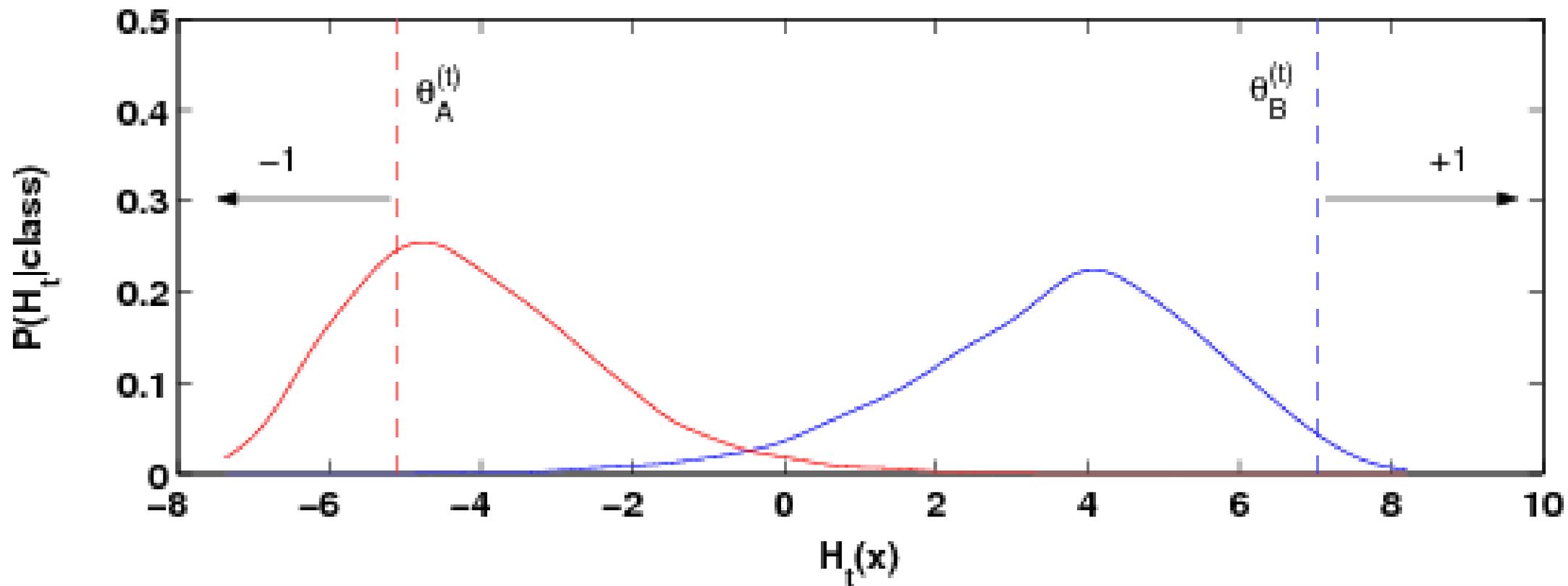




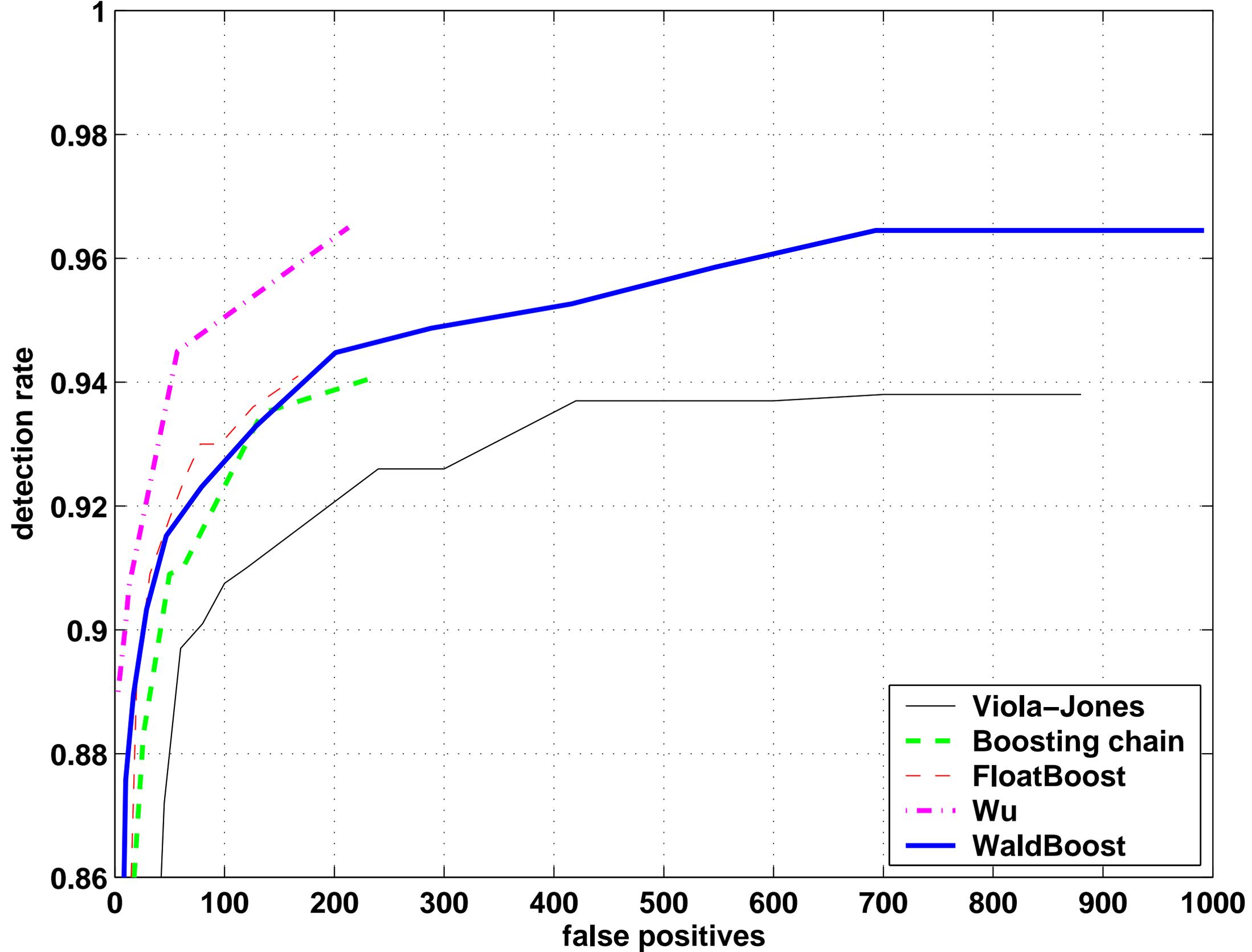




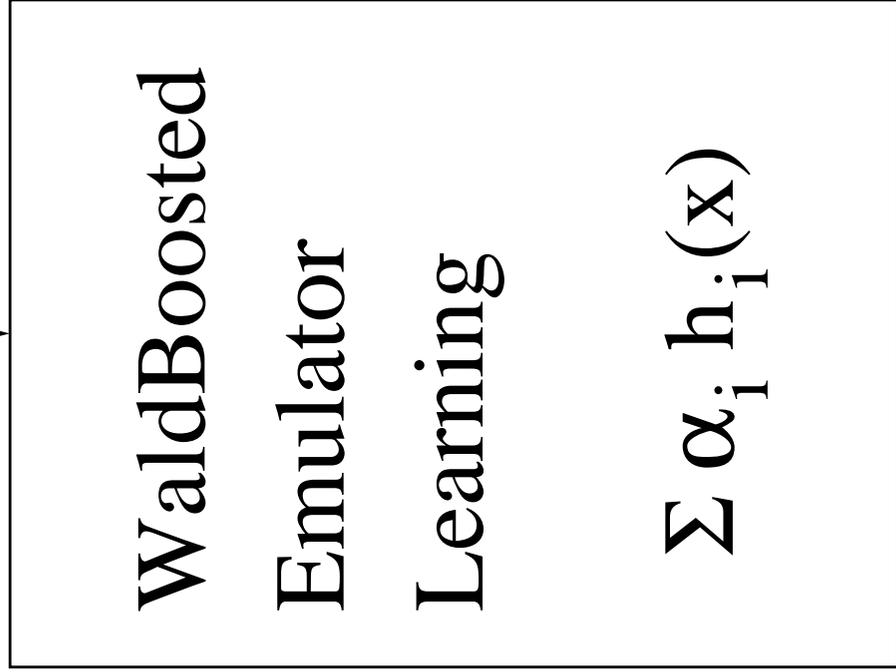
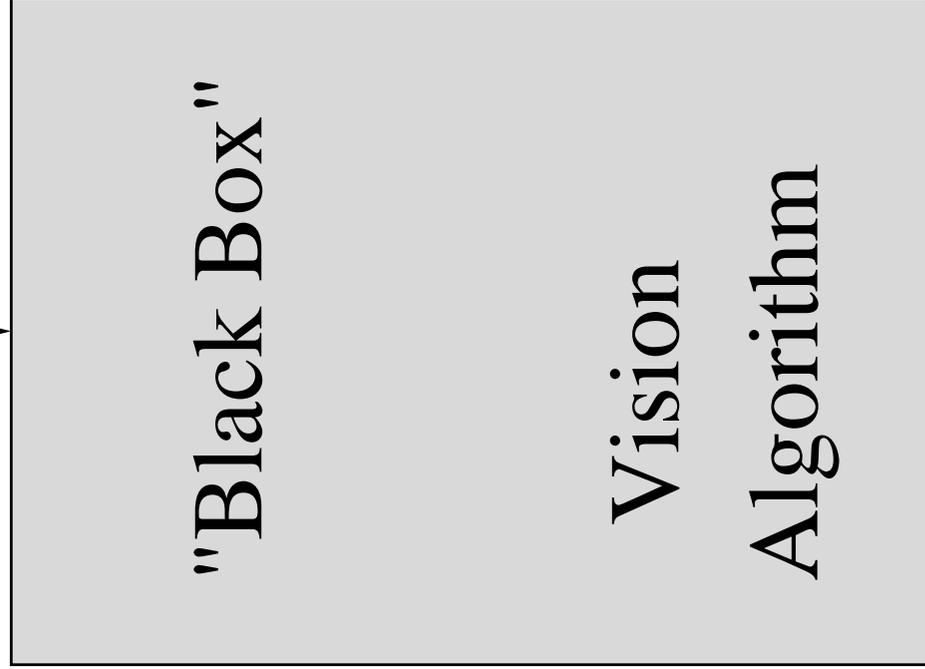
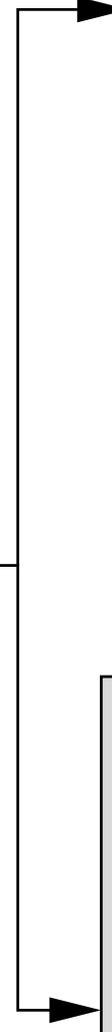


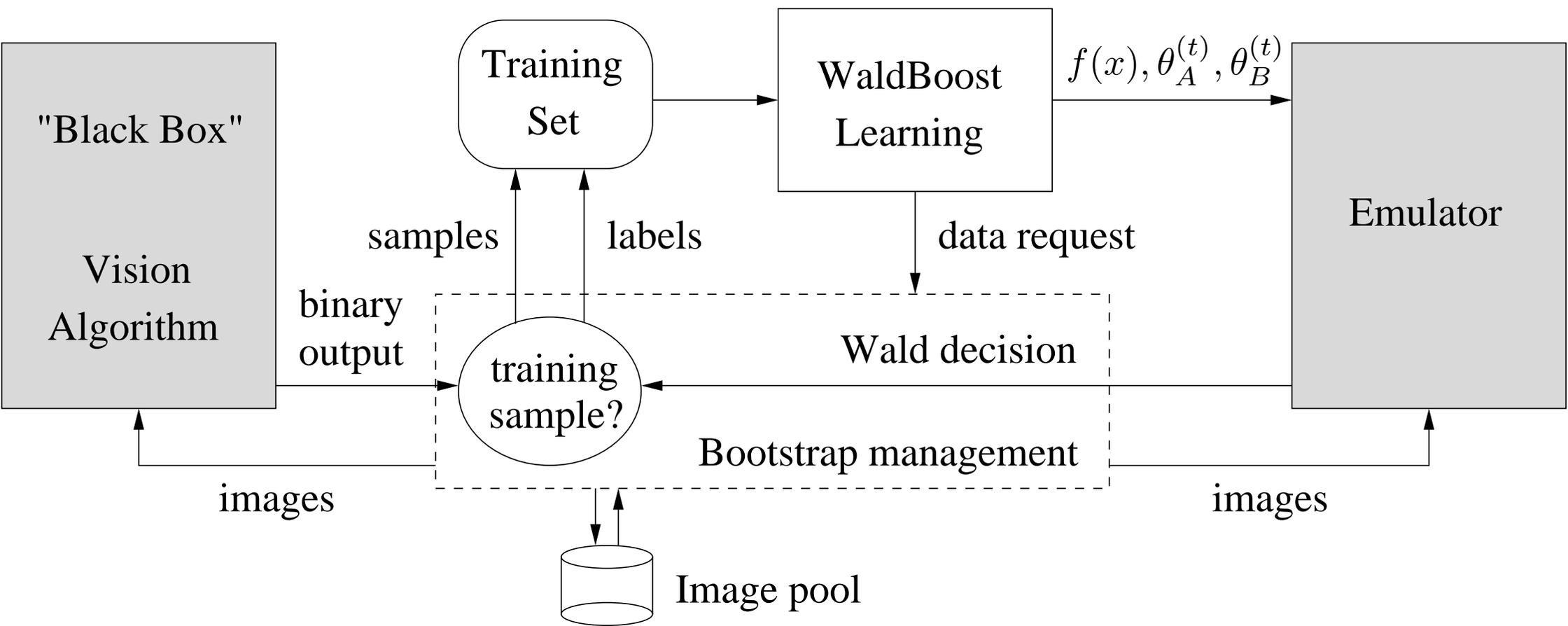


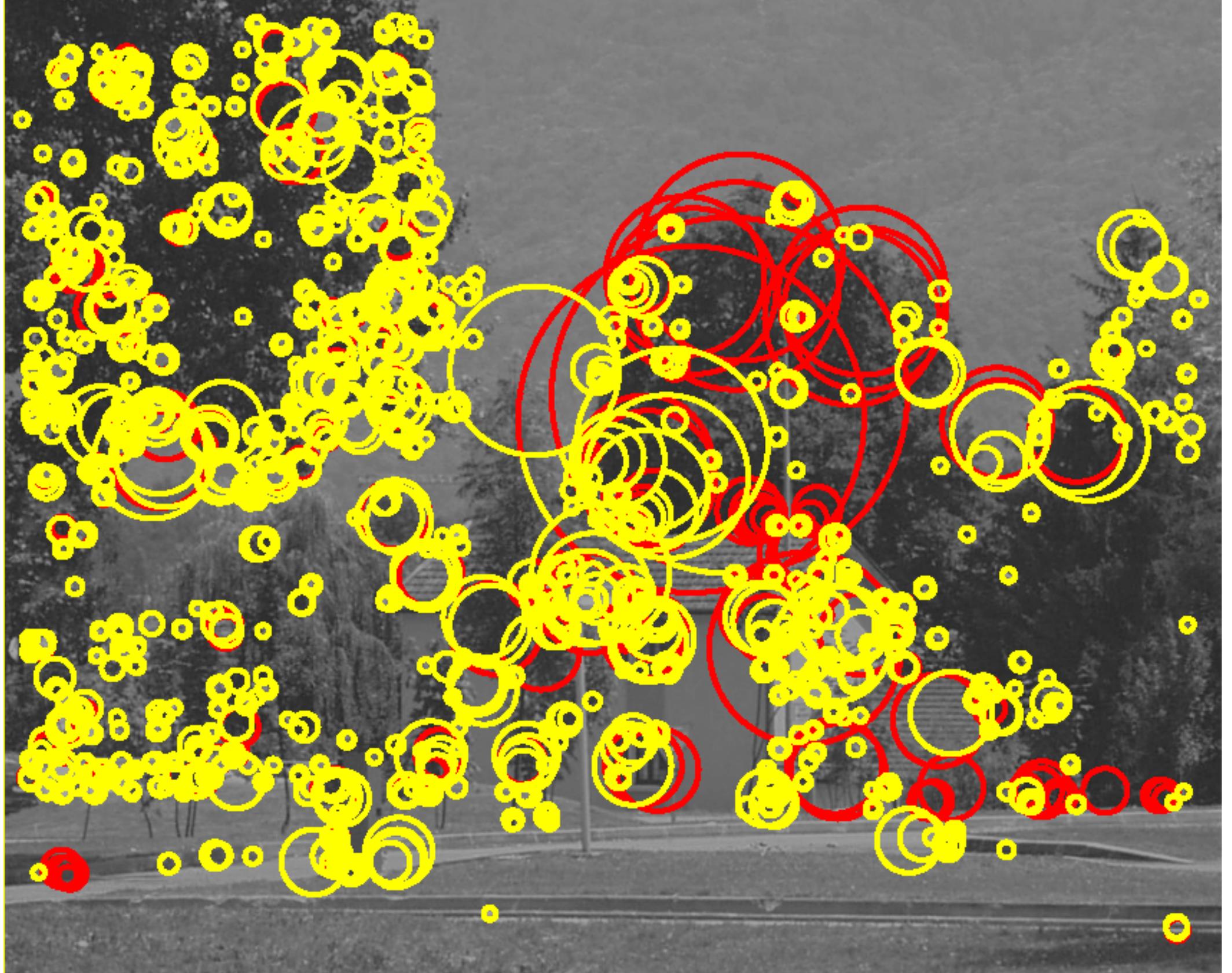


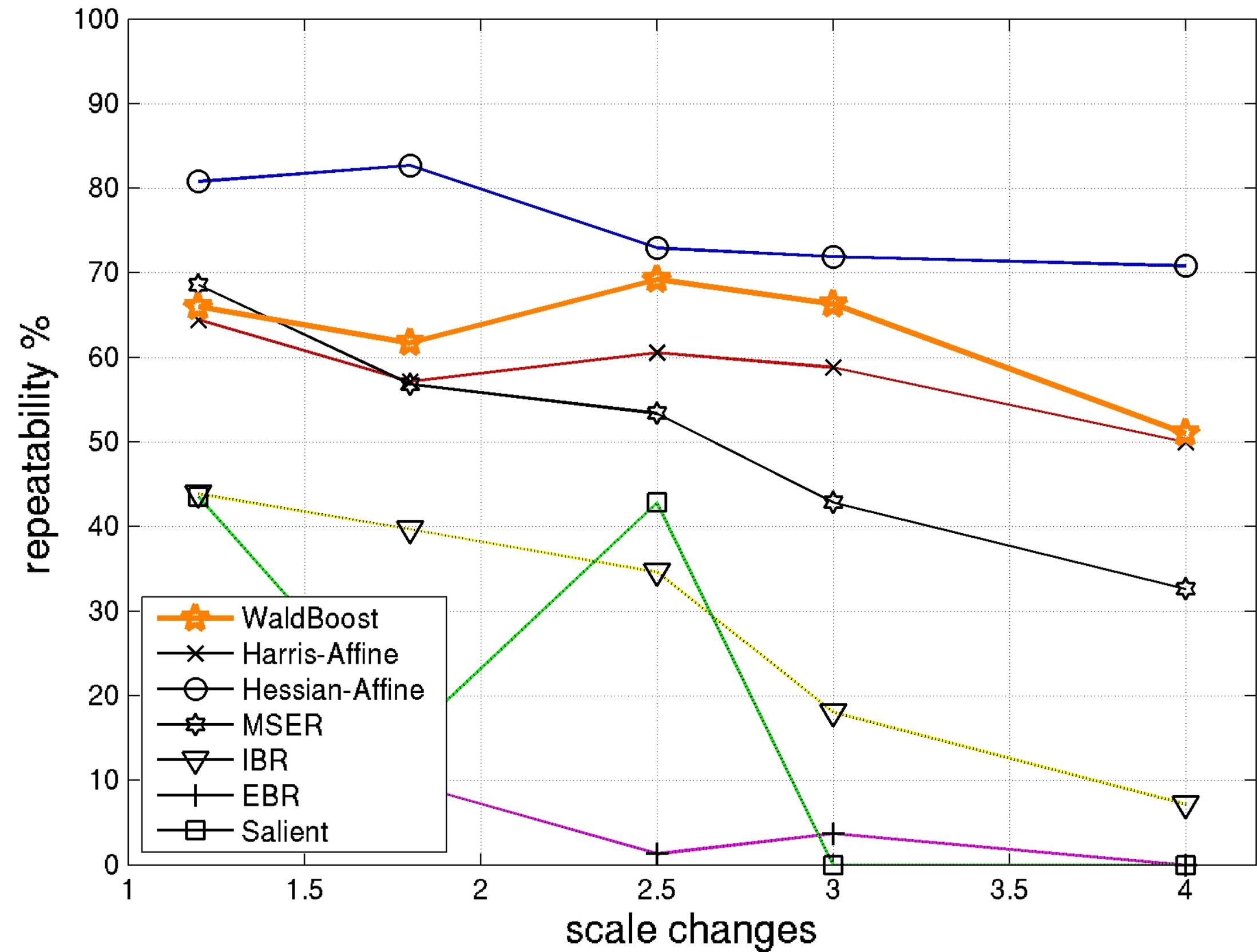


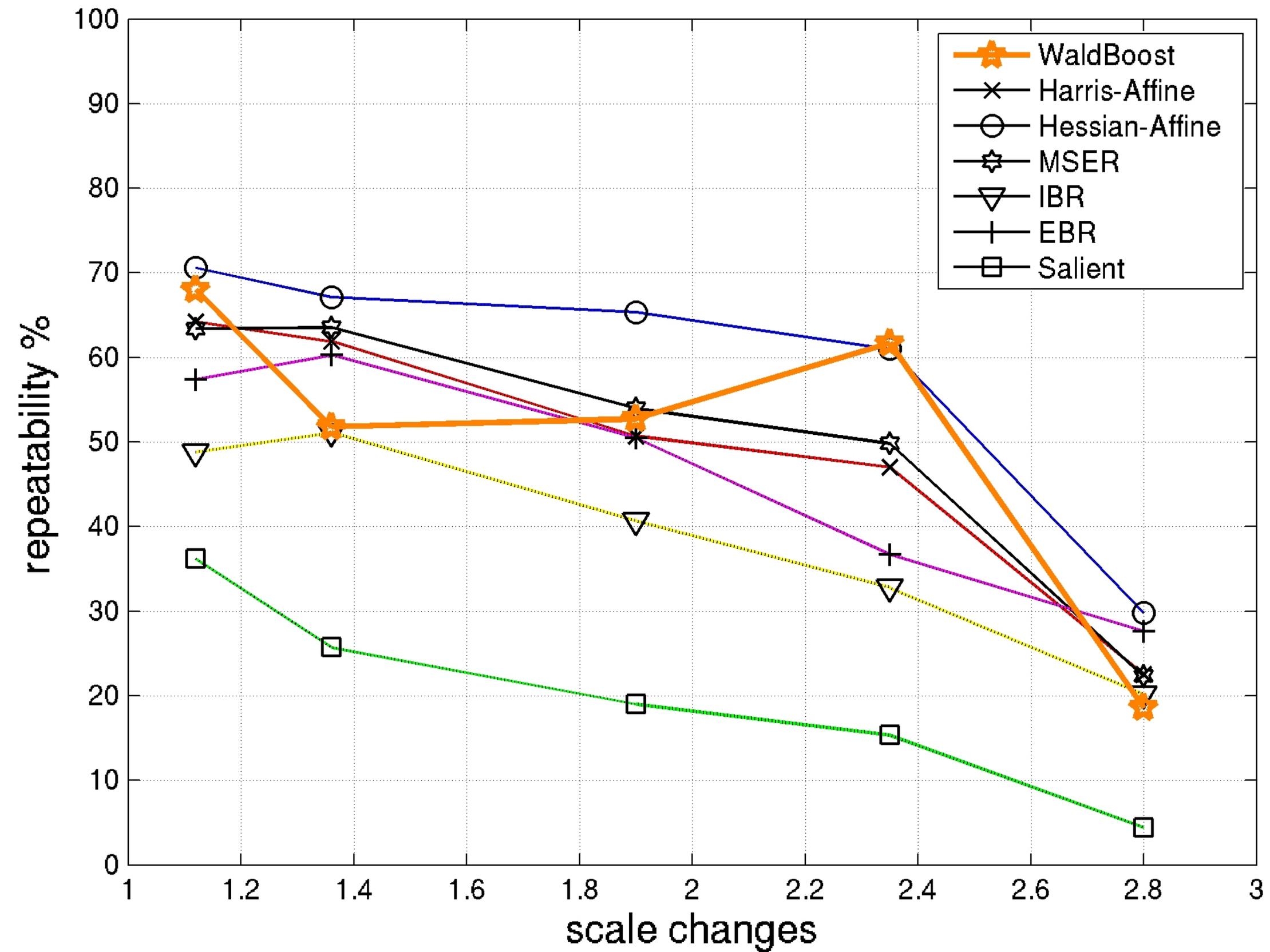
Image

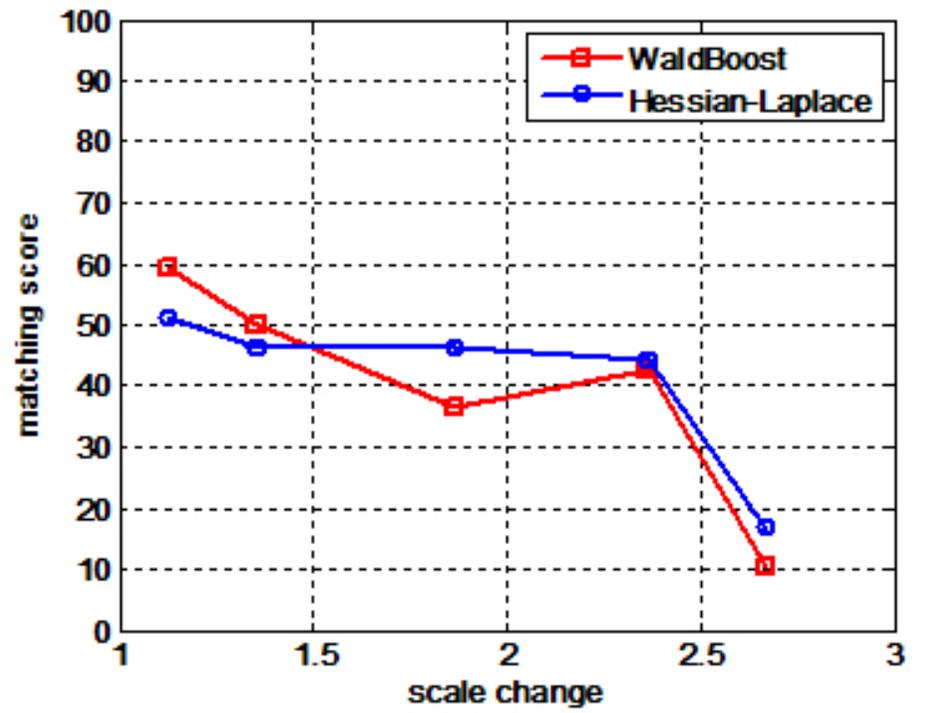
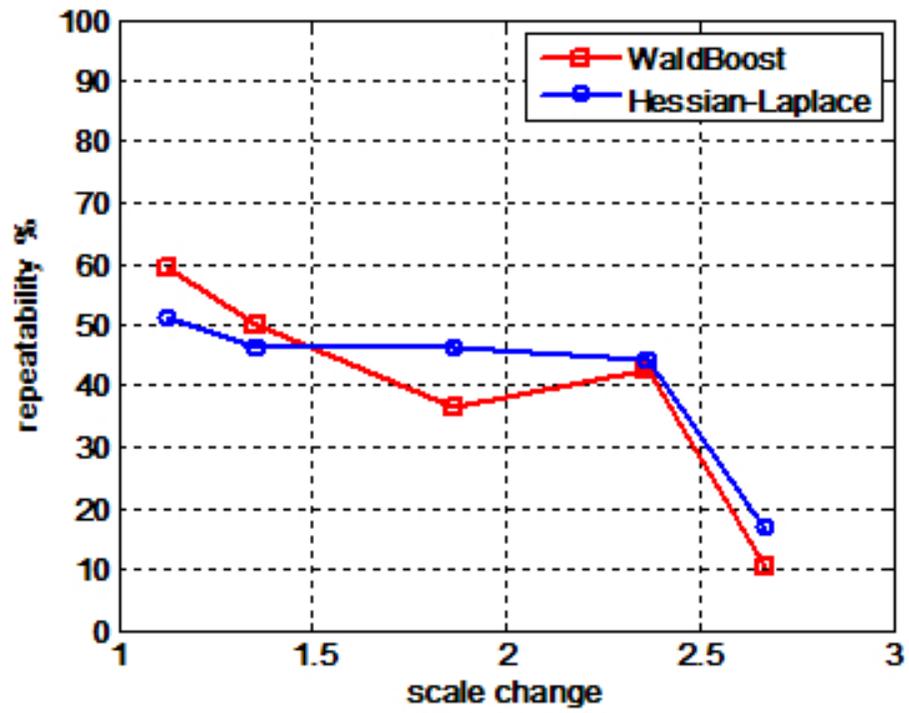


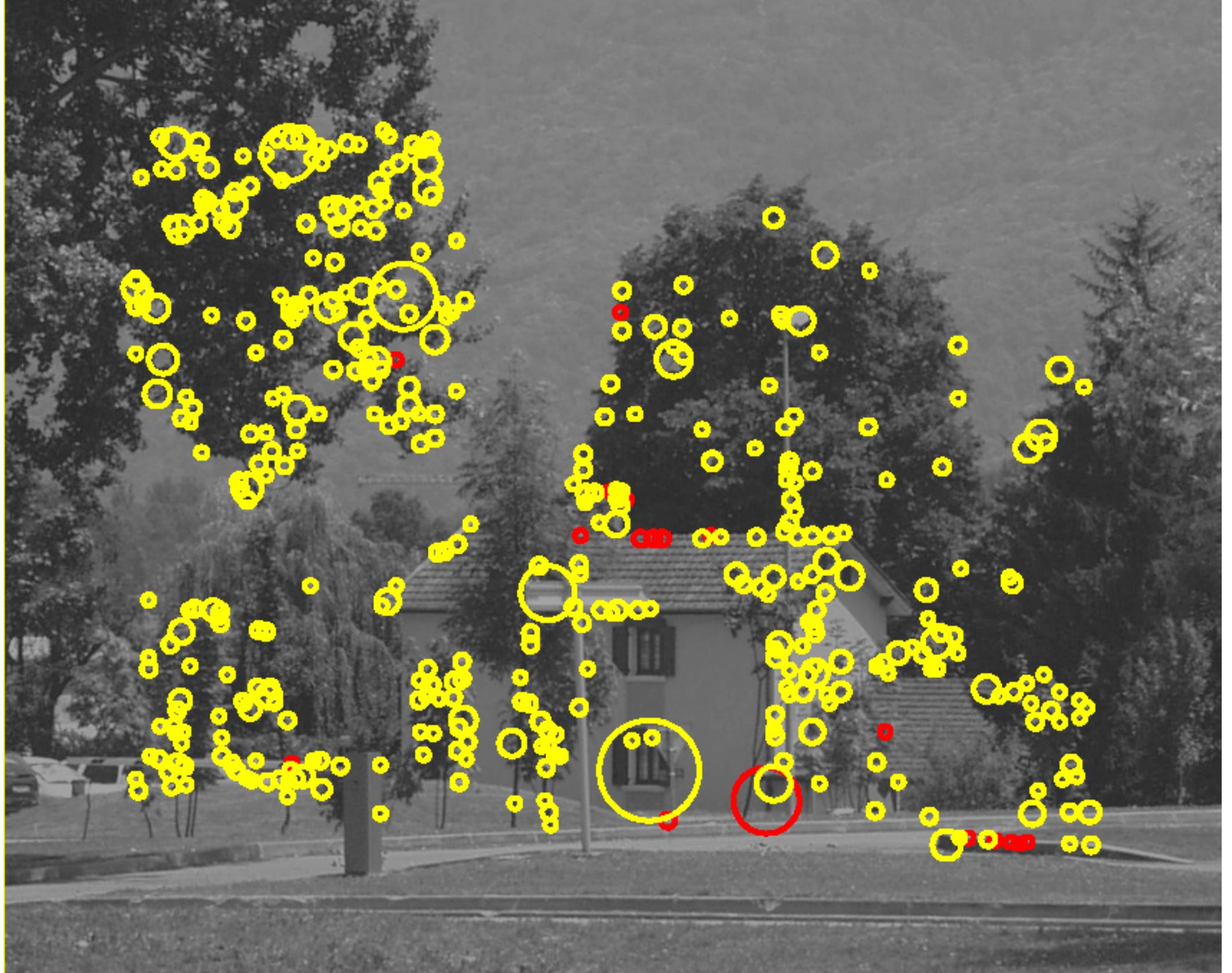


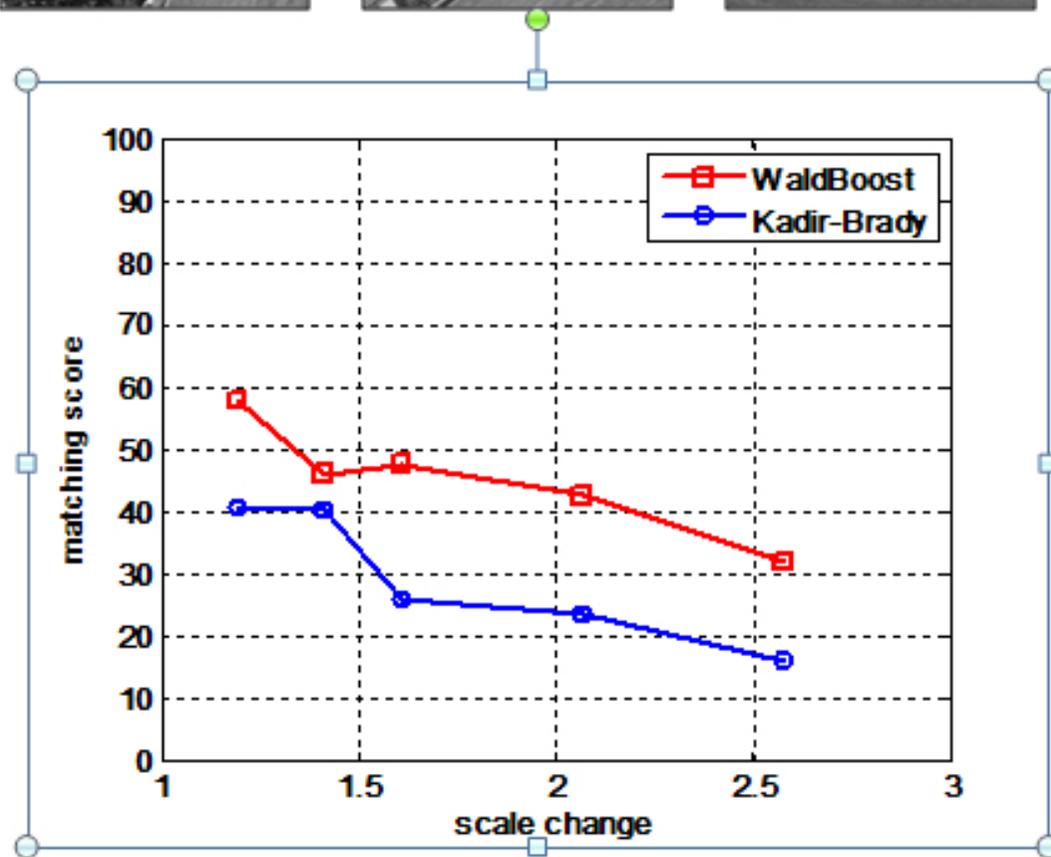
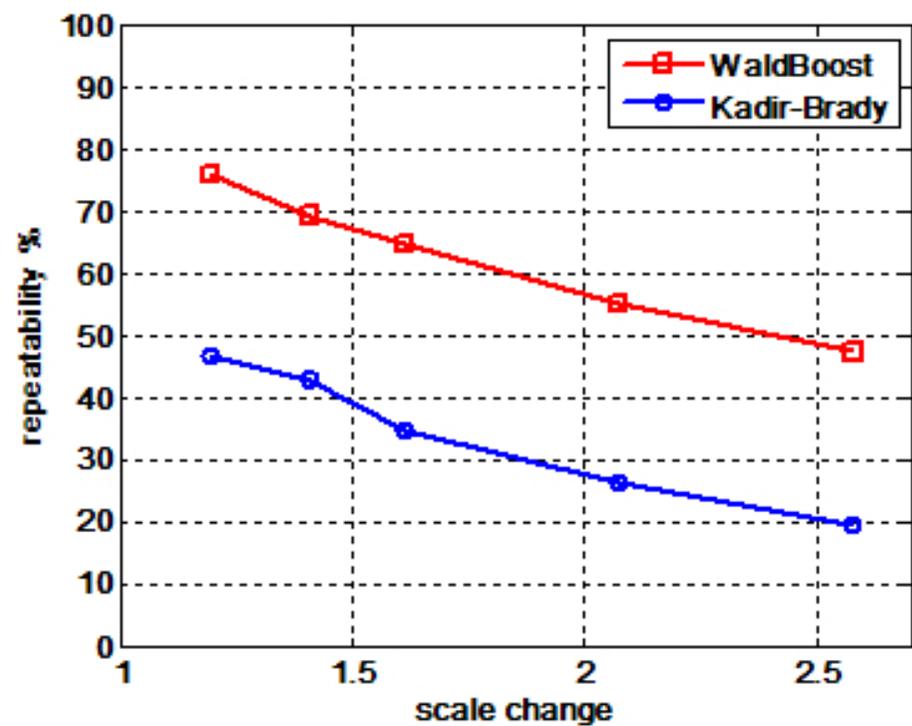
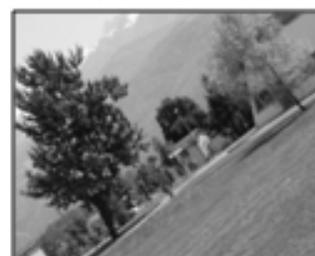
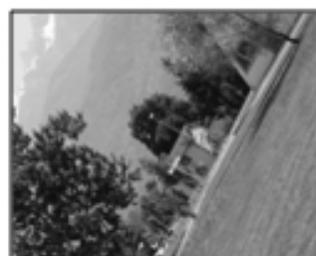


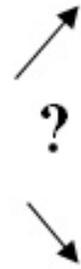


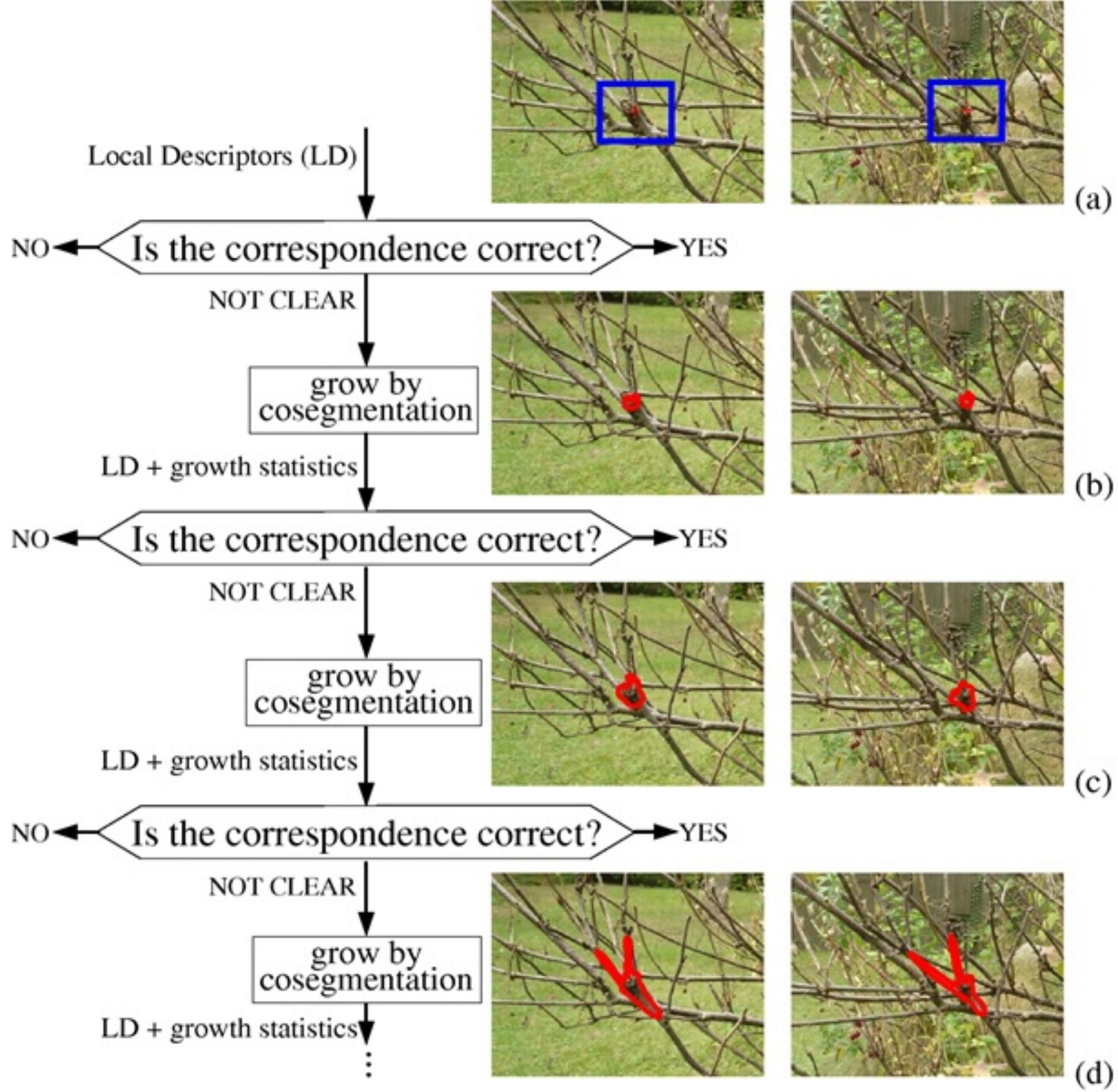


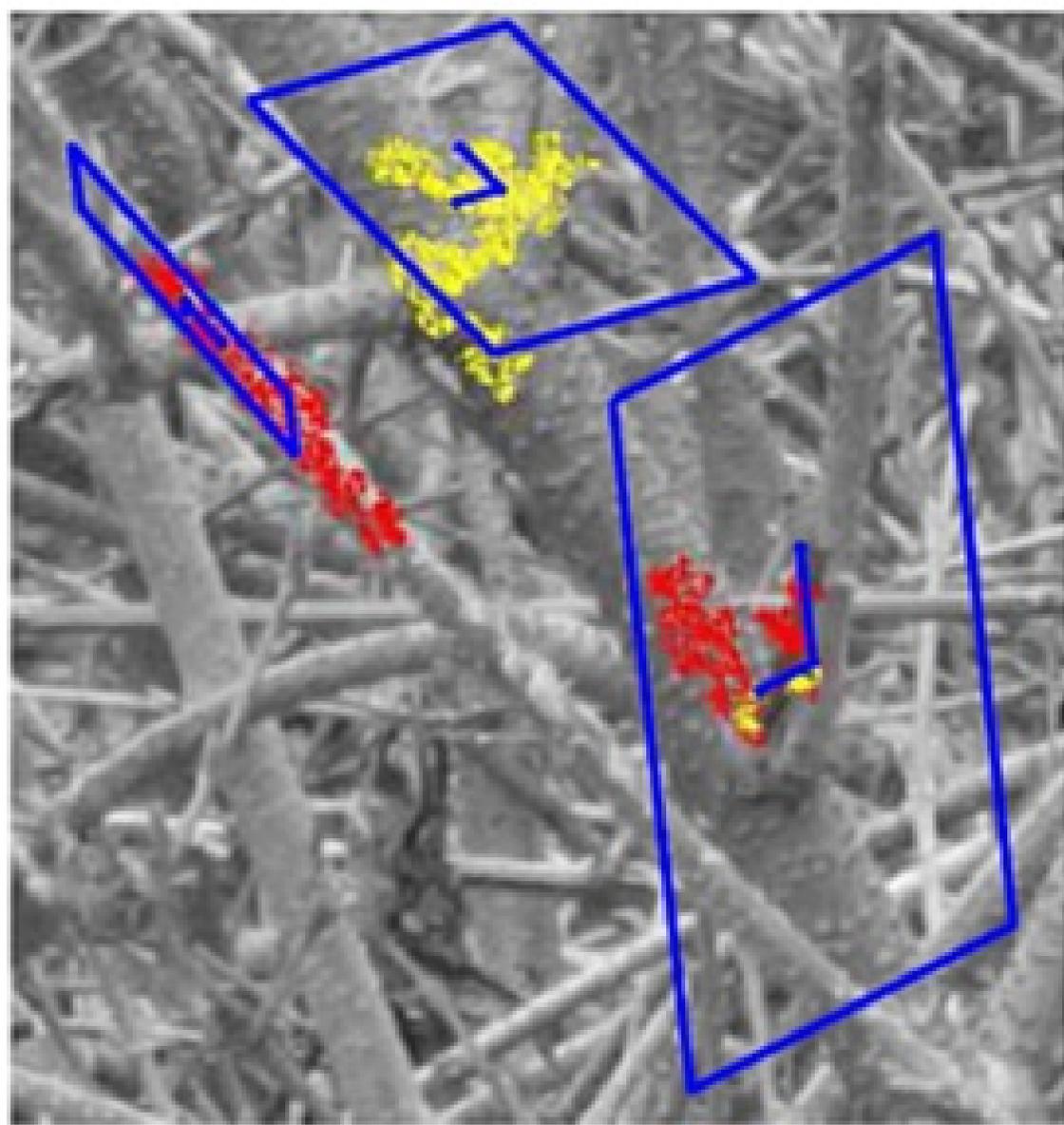
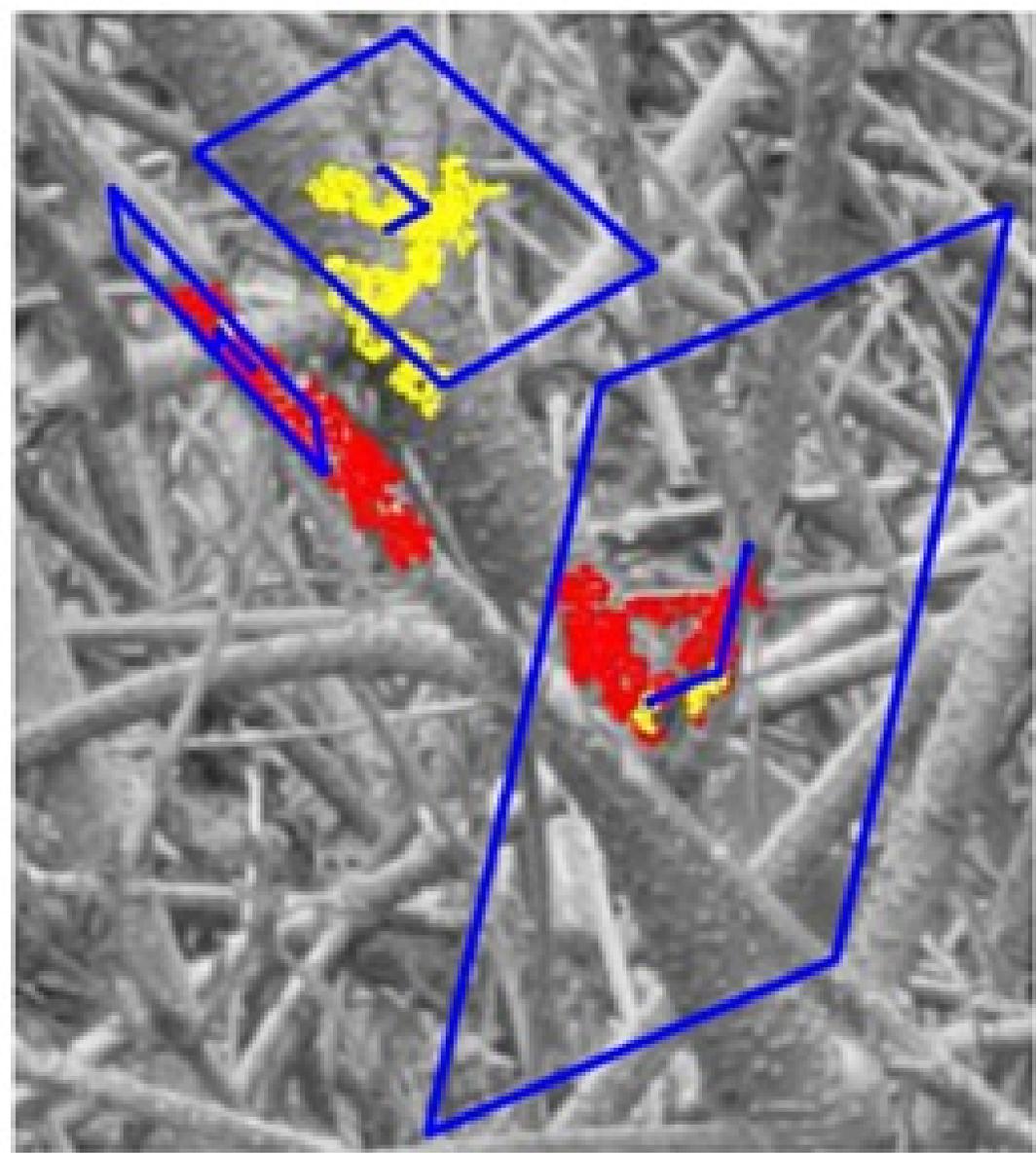


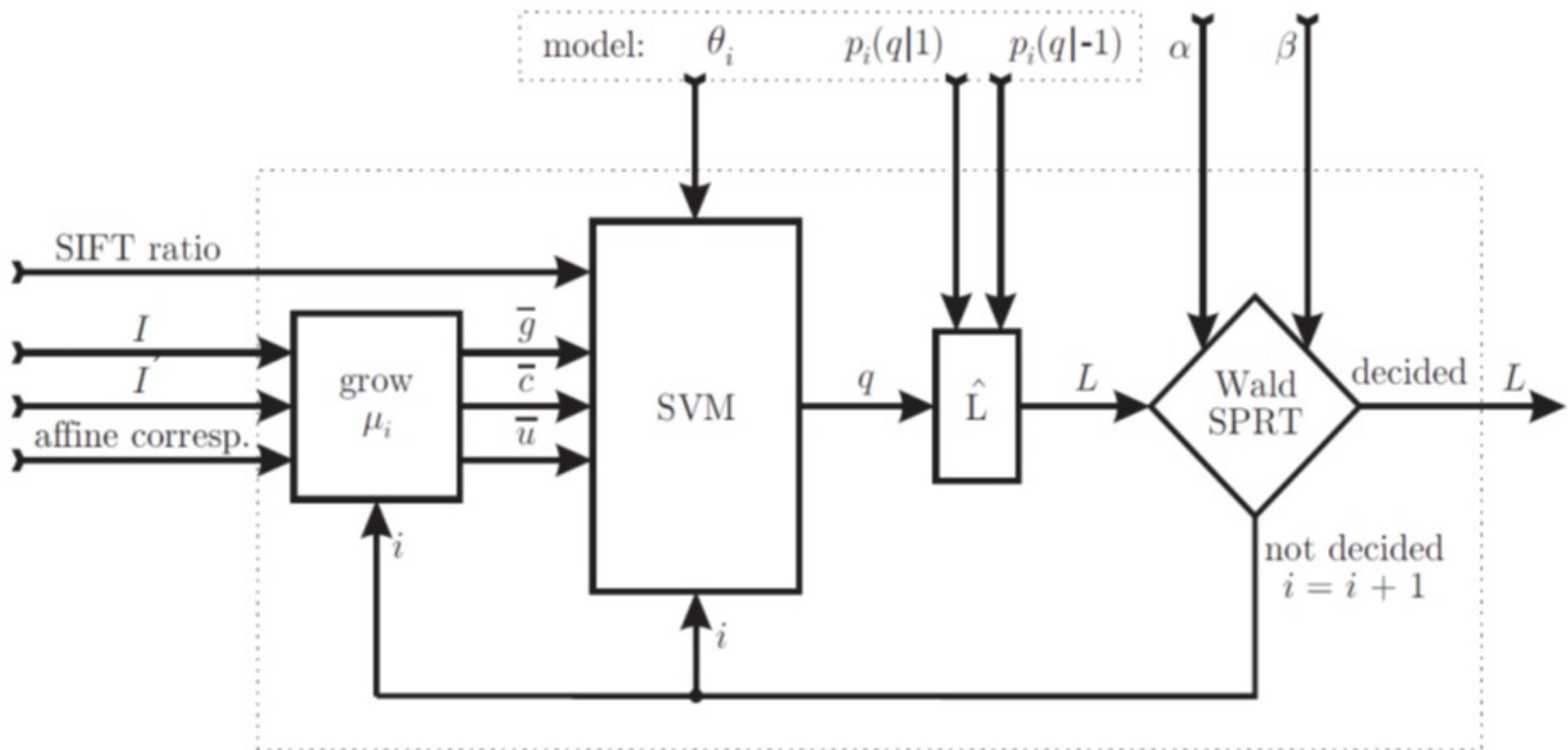




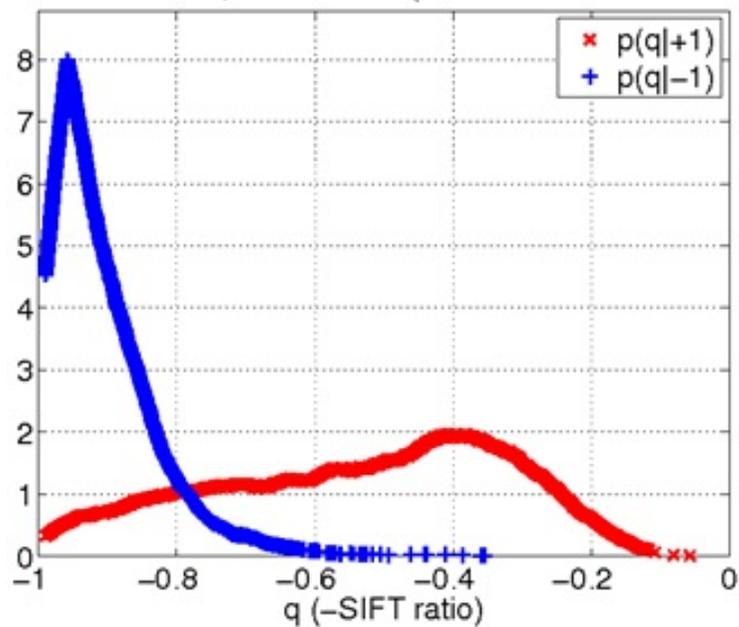




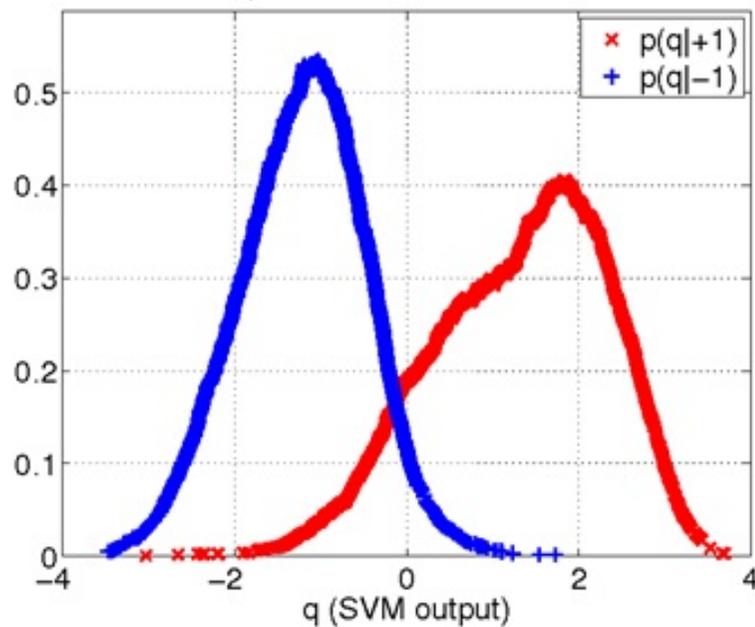




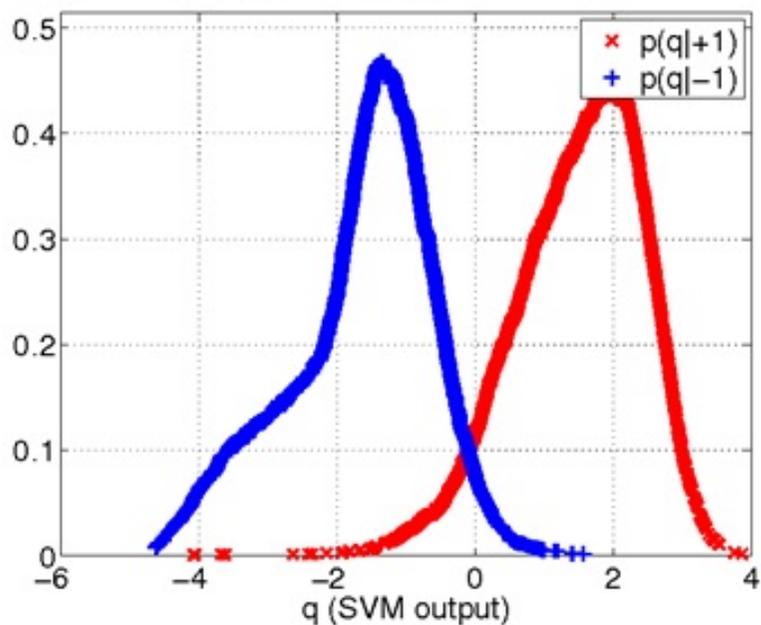
$i = 1 : \mu = 0$ (SIFT ratio only)



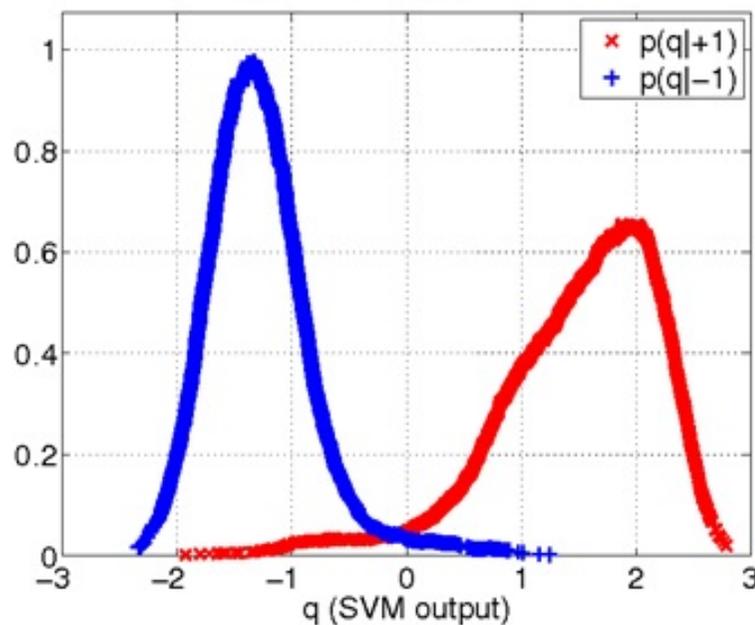
$i = 2 : \mu = 10$

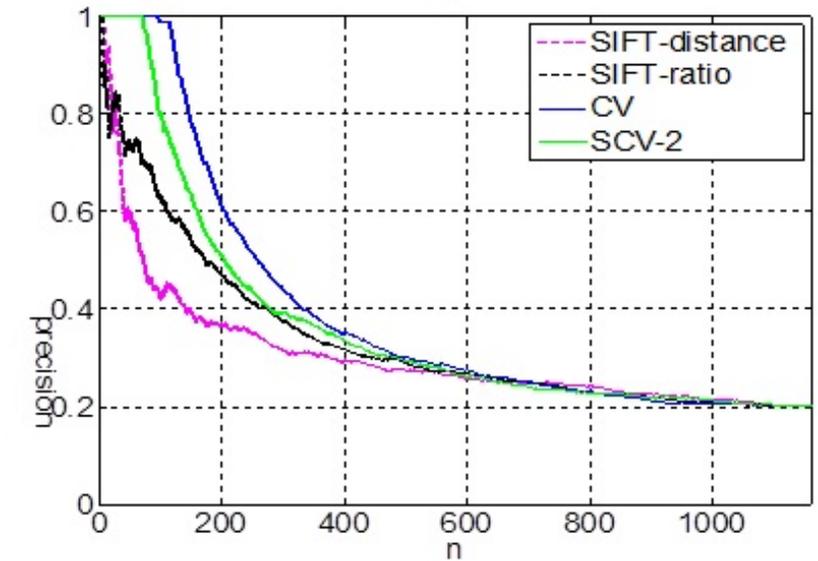
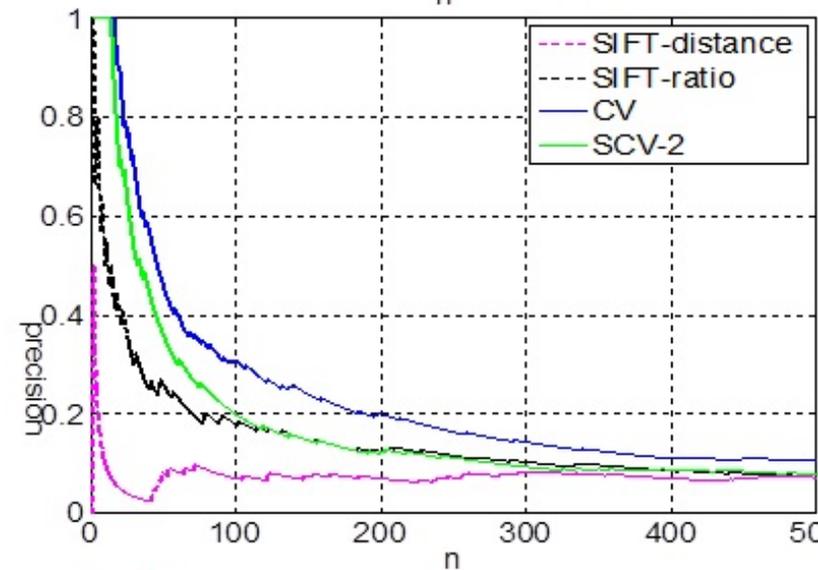
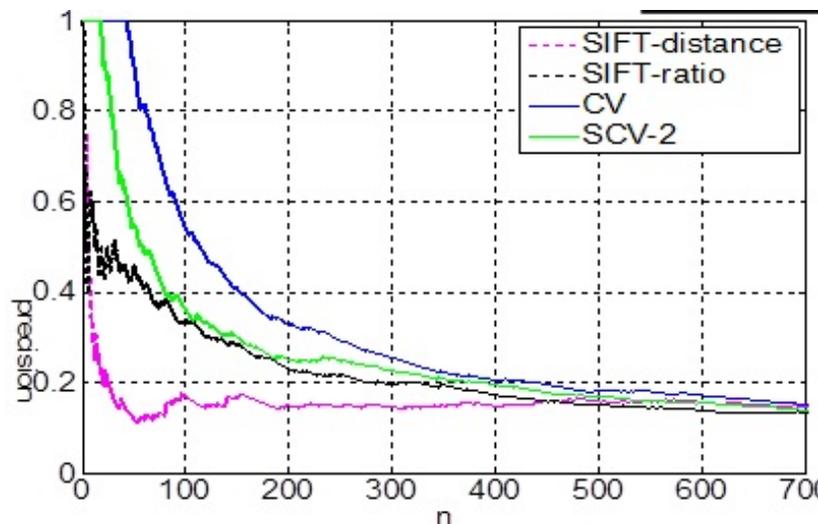
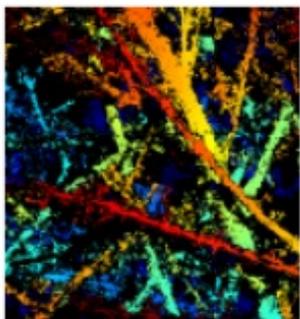


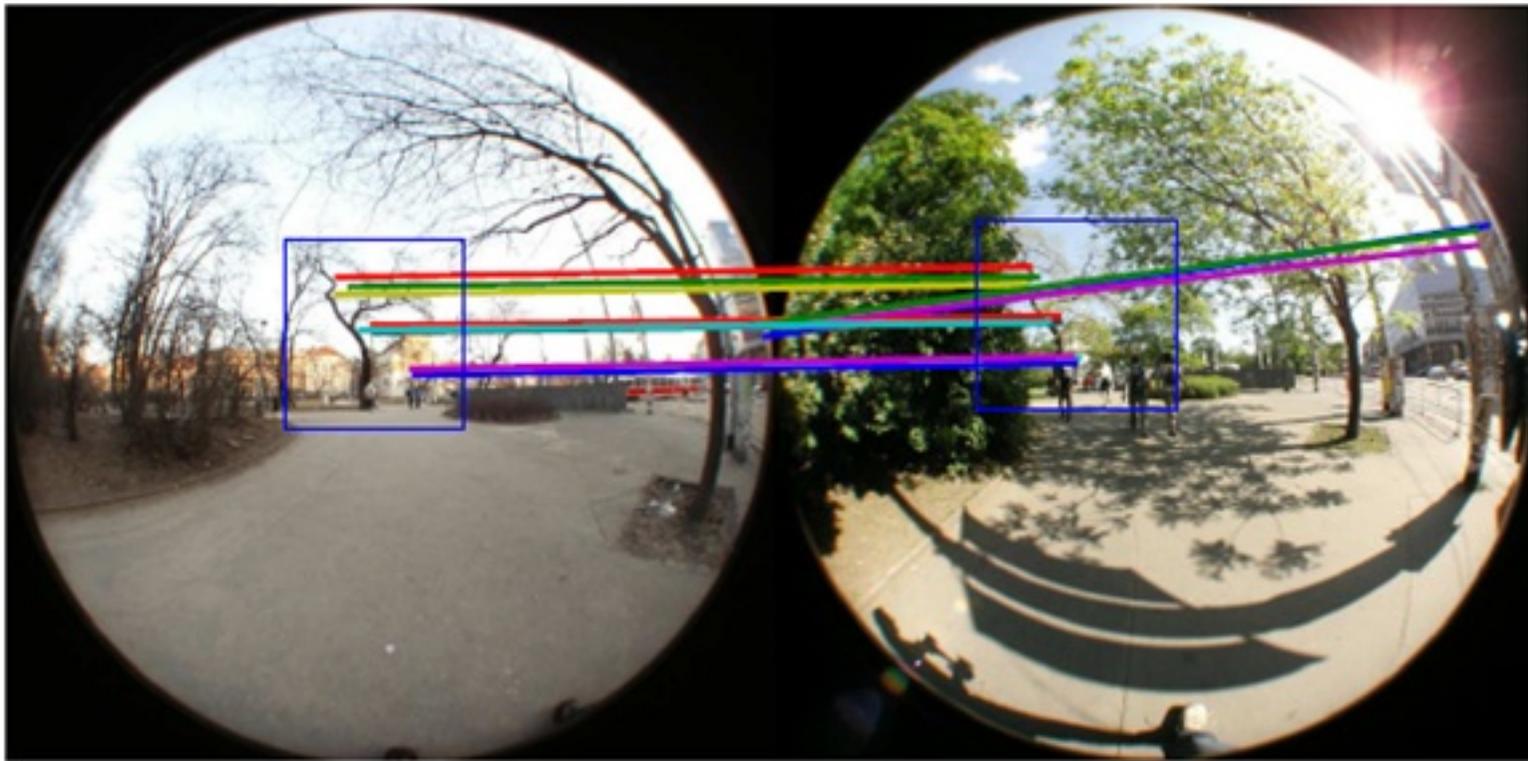
$i = 3 : \mu = 100$



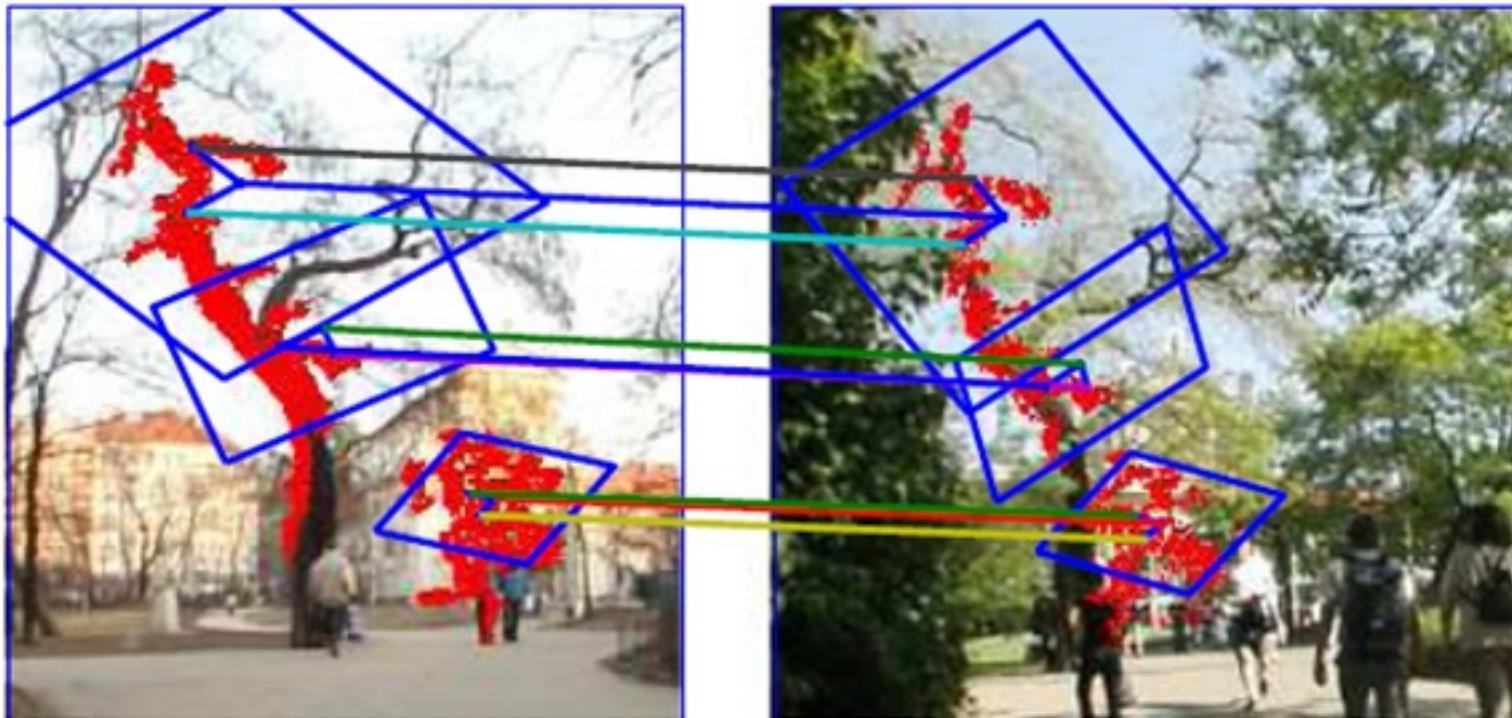
$i = 4 : \mu = 1000$







zoom (grown regions)



Repeat $k/(1-\alpha)$ times

1. Hypothesis generation

2. Model pre-verification $T_{d,d}$ test

- Verify $d \ll N$ data points, reject
- the model if not all d data points
- are consistent with the model

3. Model verification

Verify the rest of the data points

Time
 t_M

$\bar{m}_s \cdot V$

V – average number of data points verified

α – probability that a good model is rejected by $T_{d,d}$ test

$$t = \frac{k}{1 - \alpha} (t_M + \bar{m}_s V)$$

WaldSAC

Repeat $k/(1-1/A)$ times

1. Hypothesis generation

2. Model verification

use SPRT

Time

t_M

$\bar{m}_s \cdot C \log A$

$C \approx ((1 - \delta) \log \frac{1-\delta}{1-\varepsilon} + \delta \log \frac{\delta}{\varepsilon})^{-1}$

$$t(A) = \frac{k}{(1 - 1/A)} (t_M + \bar{m}_s C \log A)$$

In sequential statistical decision problem decision errors are traded off for time. These are two incomparable quantities, hence the constrained optimization.

In WaldSAC, decision errors cost time (more samples) and there is a single minimised quantity, time $t(A)$, a function of a single parameter A .