MACHINE LEARNING

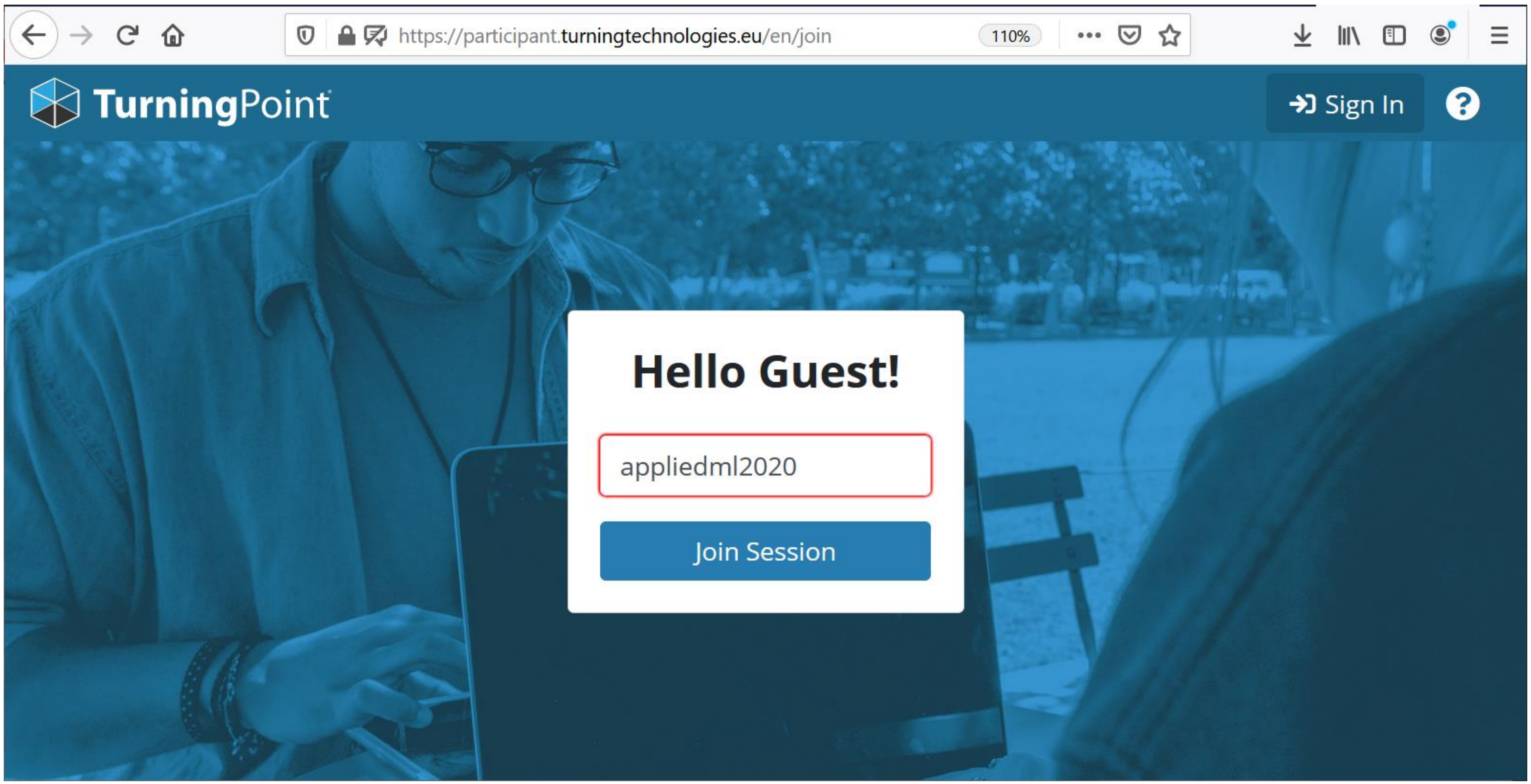
Support Vector Machine
For Classification

Interactive Lecture

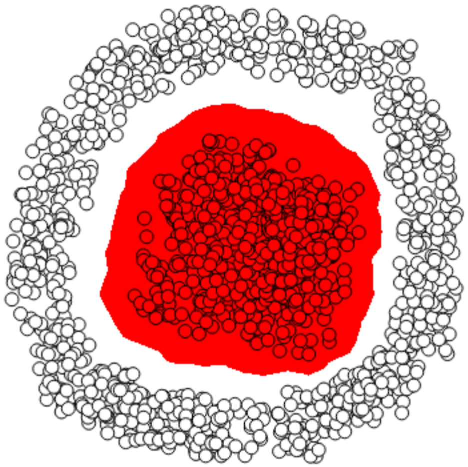
Launch polling system

https://participant.turningtechnologies.eu/en/join

Acces as GUEST and enter the session id: *appliedml2020*

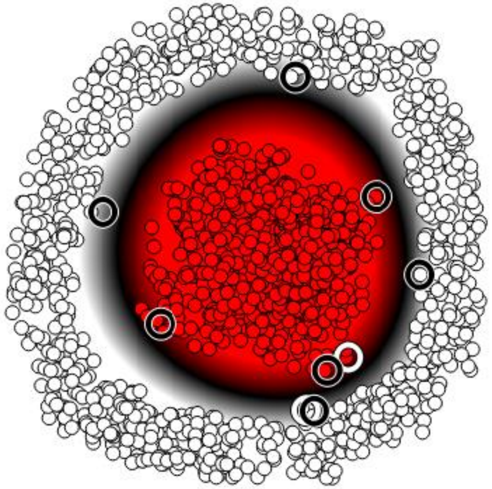


Can you recognize the classifiers?



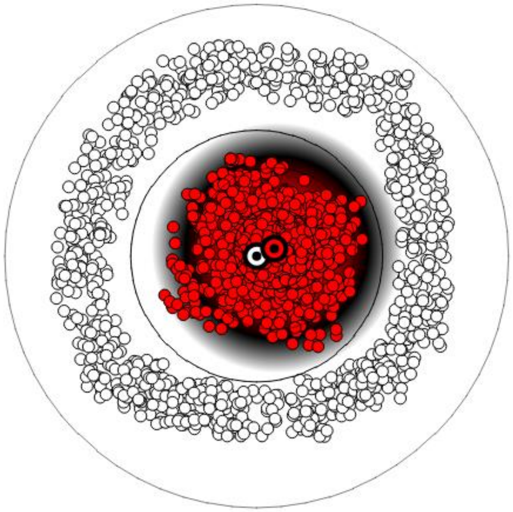
?

KNN



?

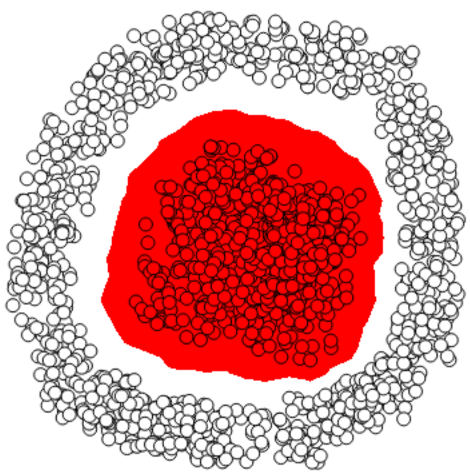
SVM



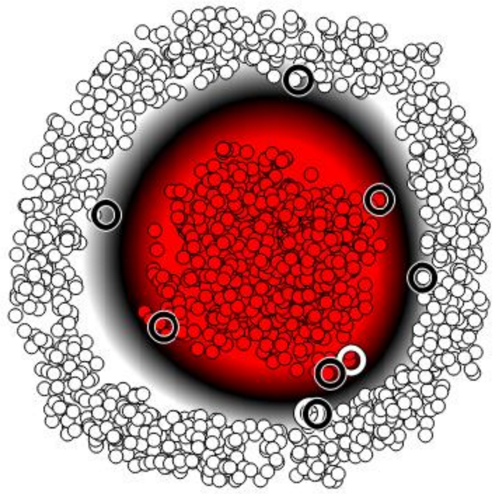
?

GMM

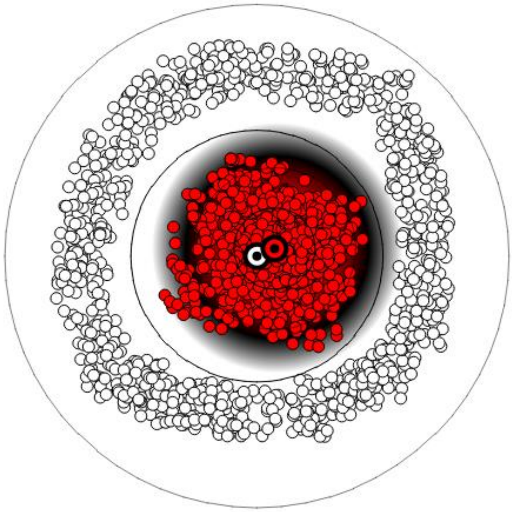
Which of the three solutions requires the least number of parameters for prediction at testing time?



KNN

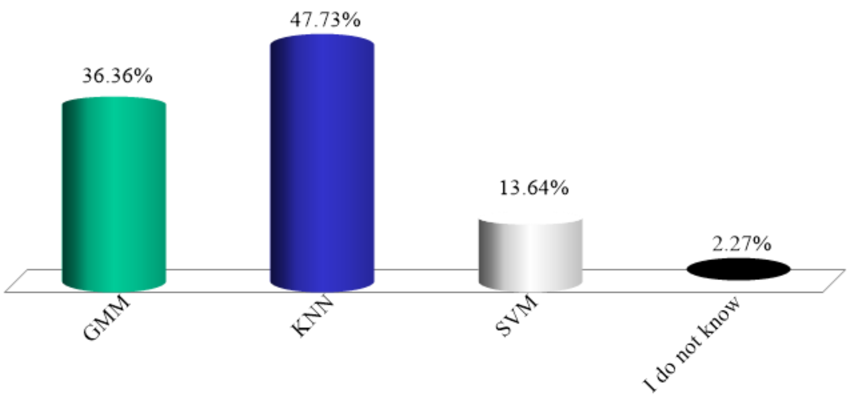


SVM

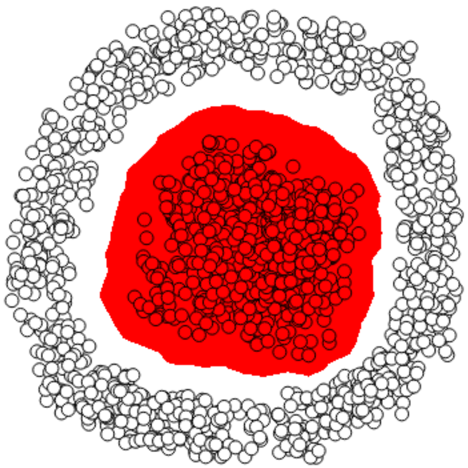


GMM

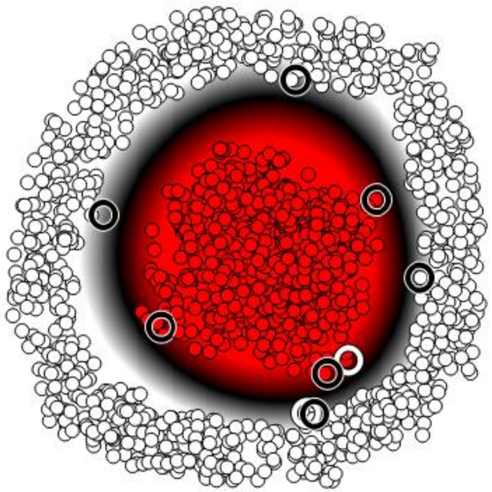
A. GMM
B. KNN
C. SVM
D. I do not know



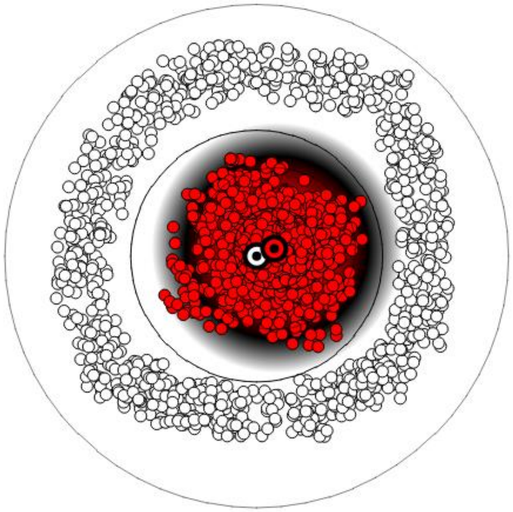
Which of the three solutions requires the least number of parameters for prediction at testing time?



KNN



SVM



GMM

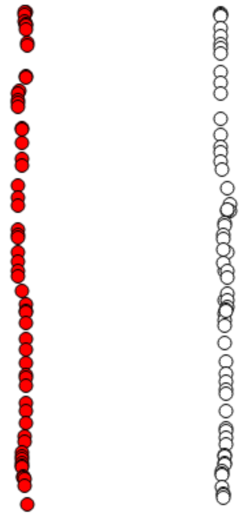
KNN is the most costly as it requires to keep all Datapoints at testing.

SVM requires 9 SVs, each 2D
→ 9*2=18 parameters
→ + 8 parameters for the α
→ +1 for b

$$f(x) = sgn\left(\sum_{i=1}^{M} \alpha_i y^i k(x, x^i) + b\right)$$

GMM is the least costly as it requires 2 spherical Gauss fcts. Each fct is represented by its 2-dim. mean and its variance → (2+1)*2=6 parameters + 2 priors on Gauss fcts

How many Support Vectors (SV) do you need at minimum when using linear SVM?

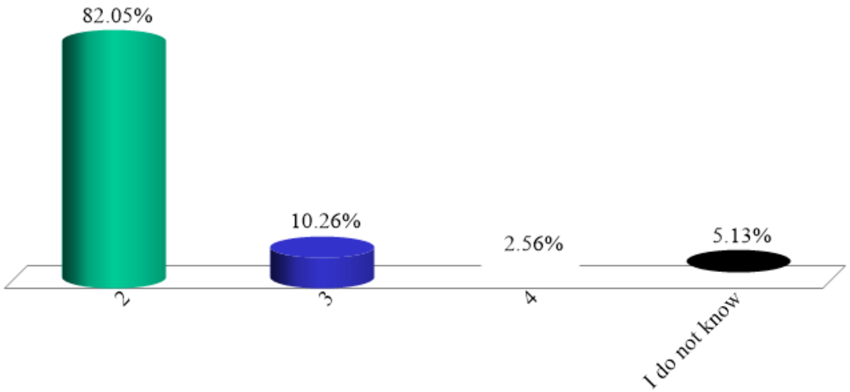


A. 2

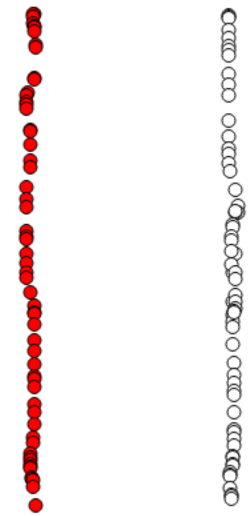
B. 3

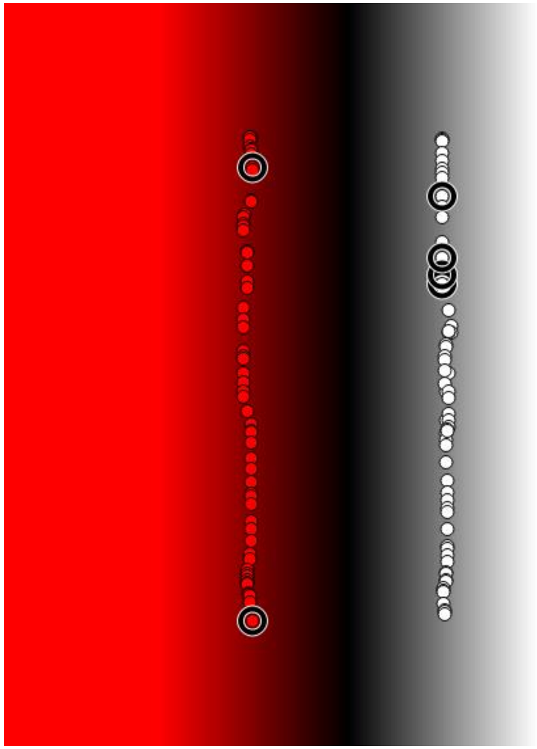
C. 4

D. I do not know



How many Support Vectors (SV) do you need at minimum when using linear SVM?



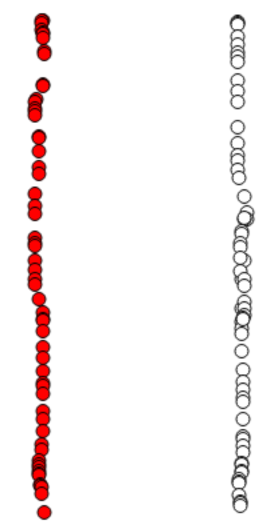


$$\sum_{i=1}^{M} \alpha_i y^i = 0$$

At minimum one needs only 2 SV-s, one in each class.

SVM usually finds more than required number of SV as the standard optimization problem is not constrained to find the minimum.

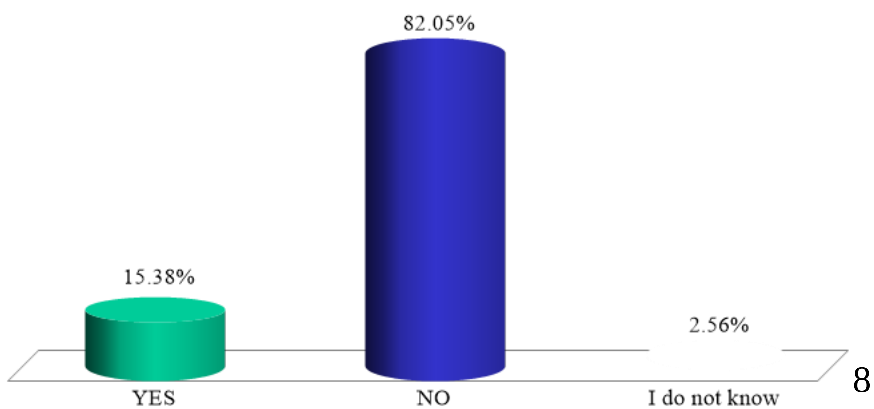
Is the solution unique?



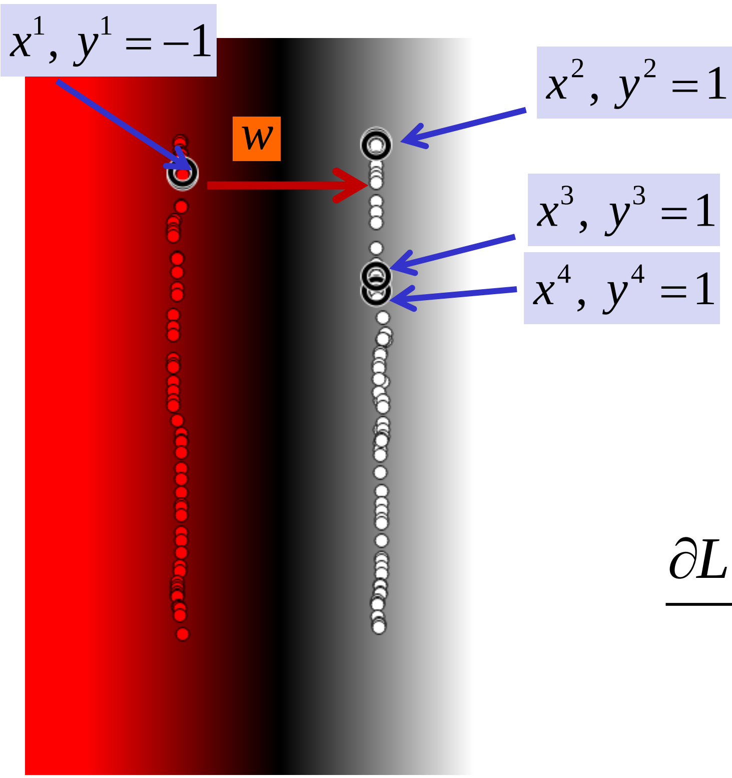
A. YES

B. NO

C. I do not know



Is the solution unique?

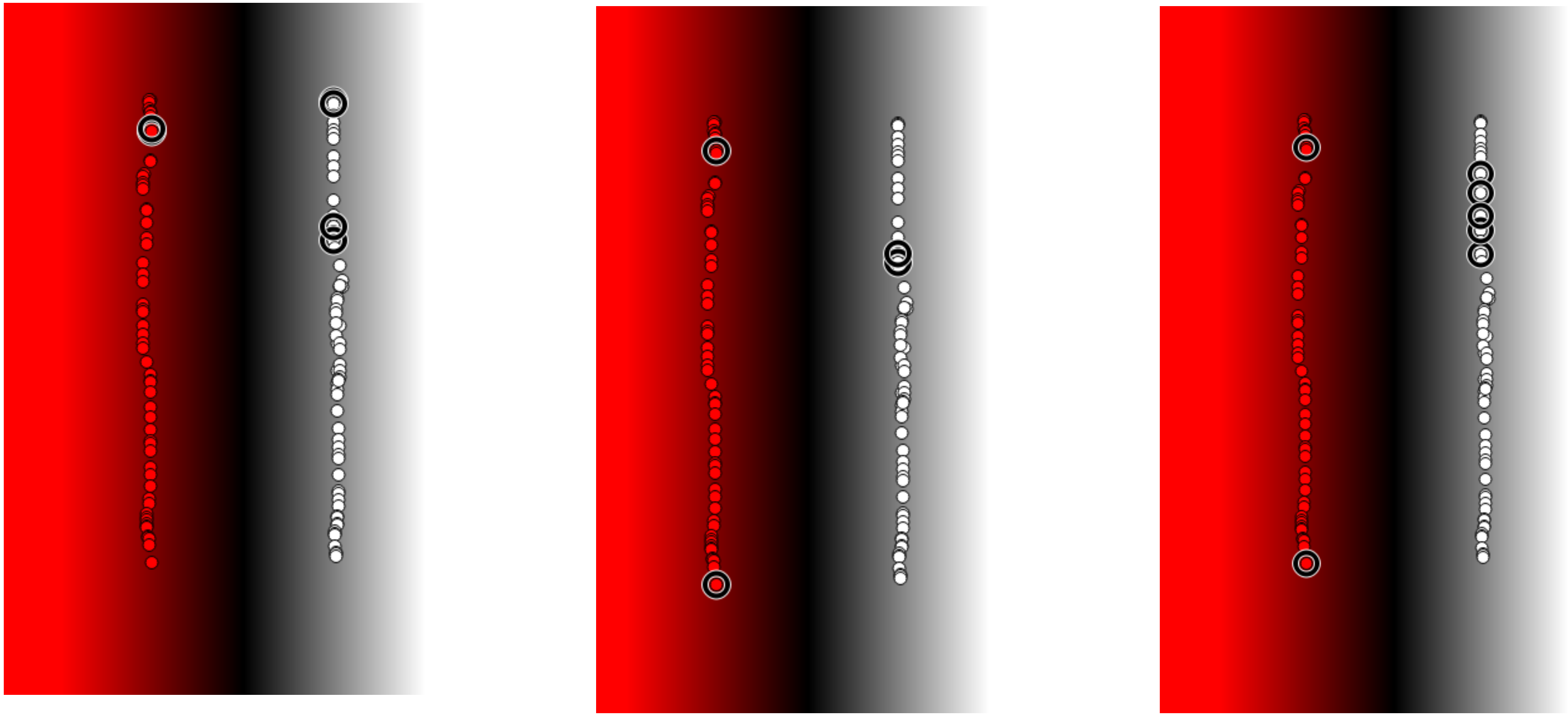


If $\alpha_1 = \alpha_2 = 1$

The vector w defines the separating hyperplane. It is determined by a linear combination of selected training points, the support vectors.

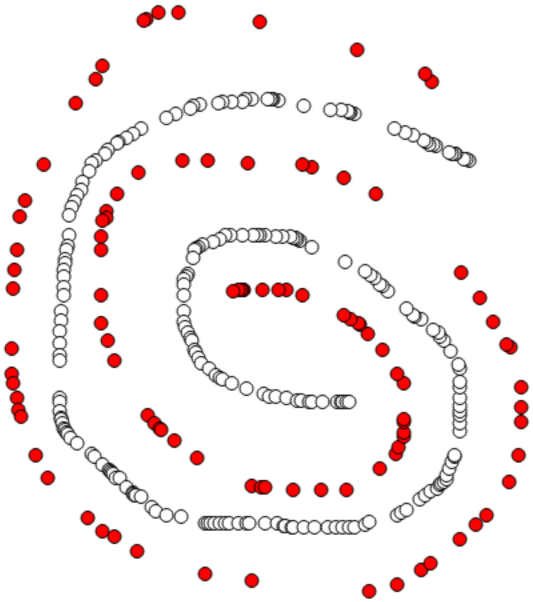
$$\frac{\partial L(w,b,\alpha)}{\partial w} = 0 \quad \Leftrightarrow \quad w = \sum_{i=1}^{M} \alpha_i y^i x^i$$

Is the solution unique?

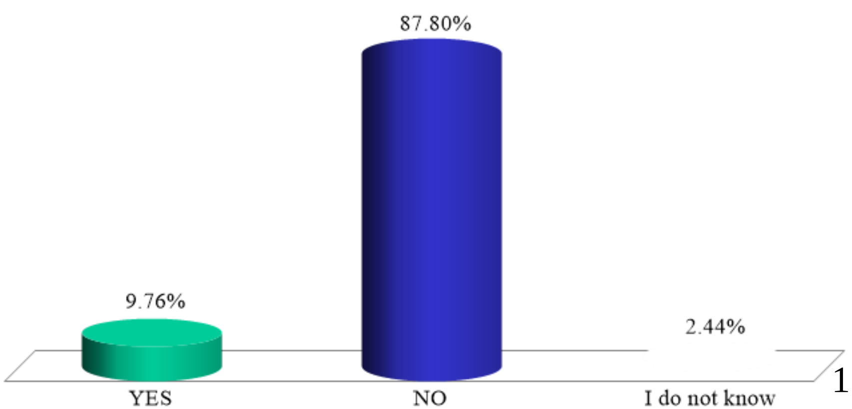


Convex optimization has a global optimum but several solutions may have the same optimum on the objective function. Here the same separating plane can be found when using different combinations of datapoints.

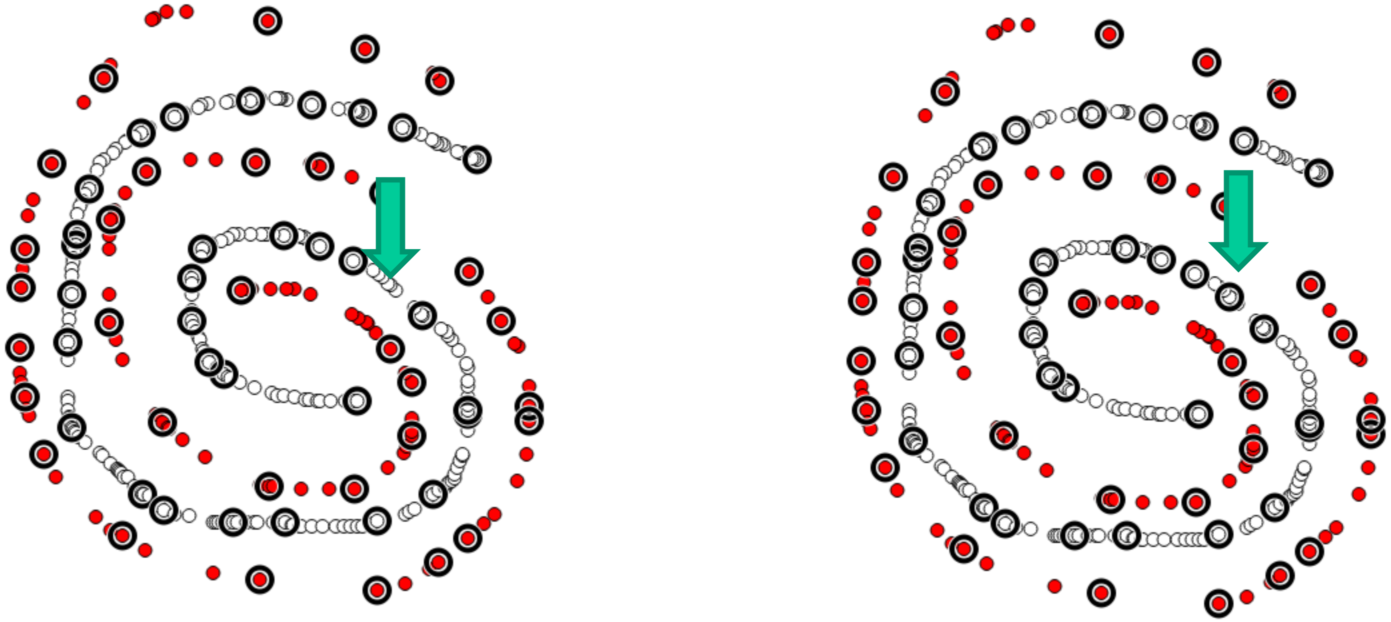
Is the solution unique?



A. YES
B. NO
C. I do not know

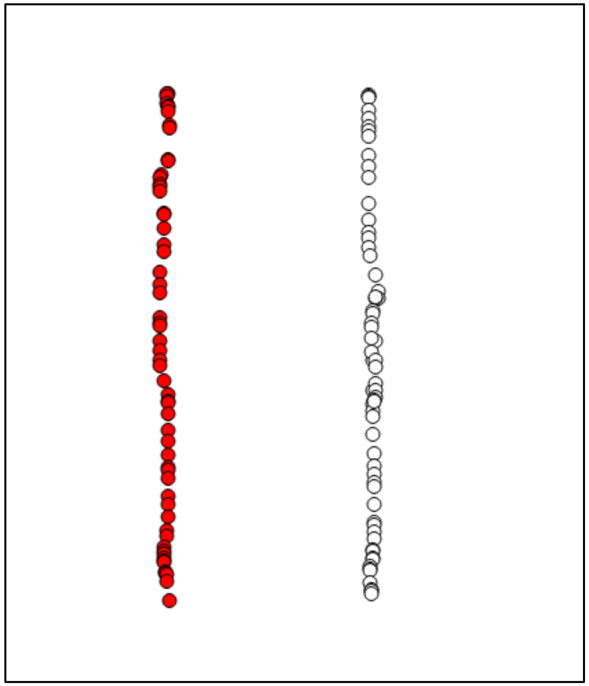


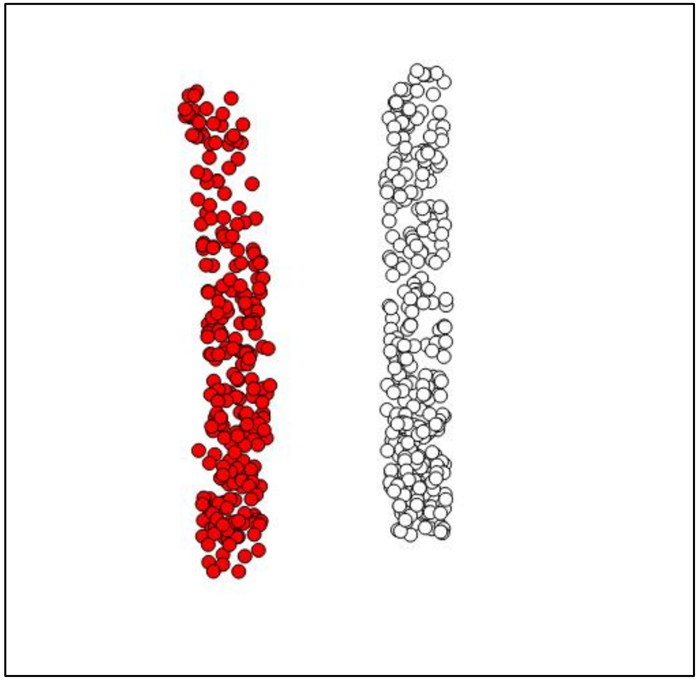
Is the solution unique?



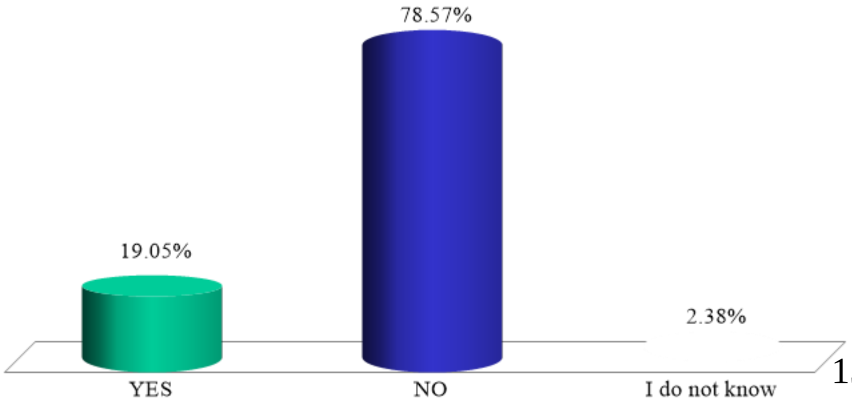
The solution is not unique either but as data become more complex, less combination will lead to global optimum. Only redundant points may be discarded from one run to the next.

Will the number of Support Vectors (SV) change as we have more datapoints?

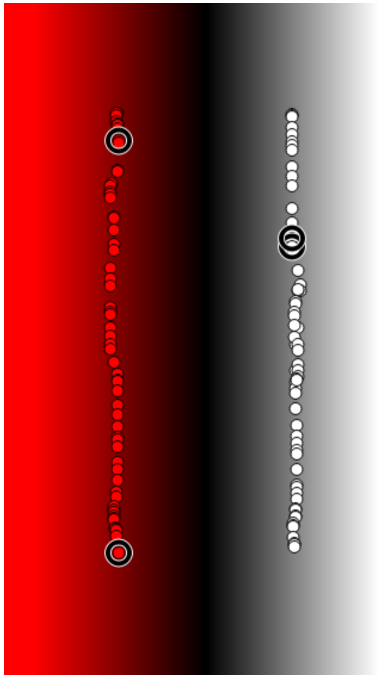


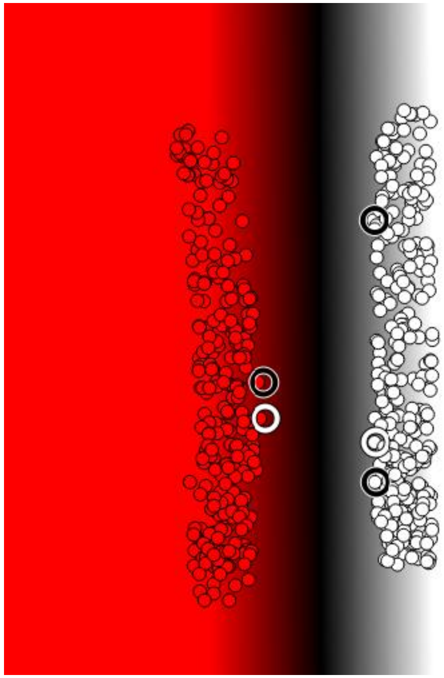


A. YES
B. NO
C. I do not know



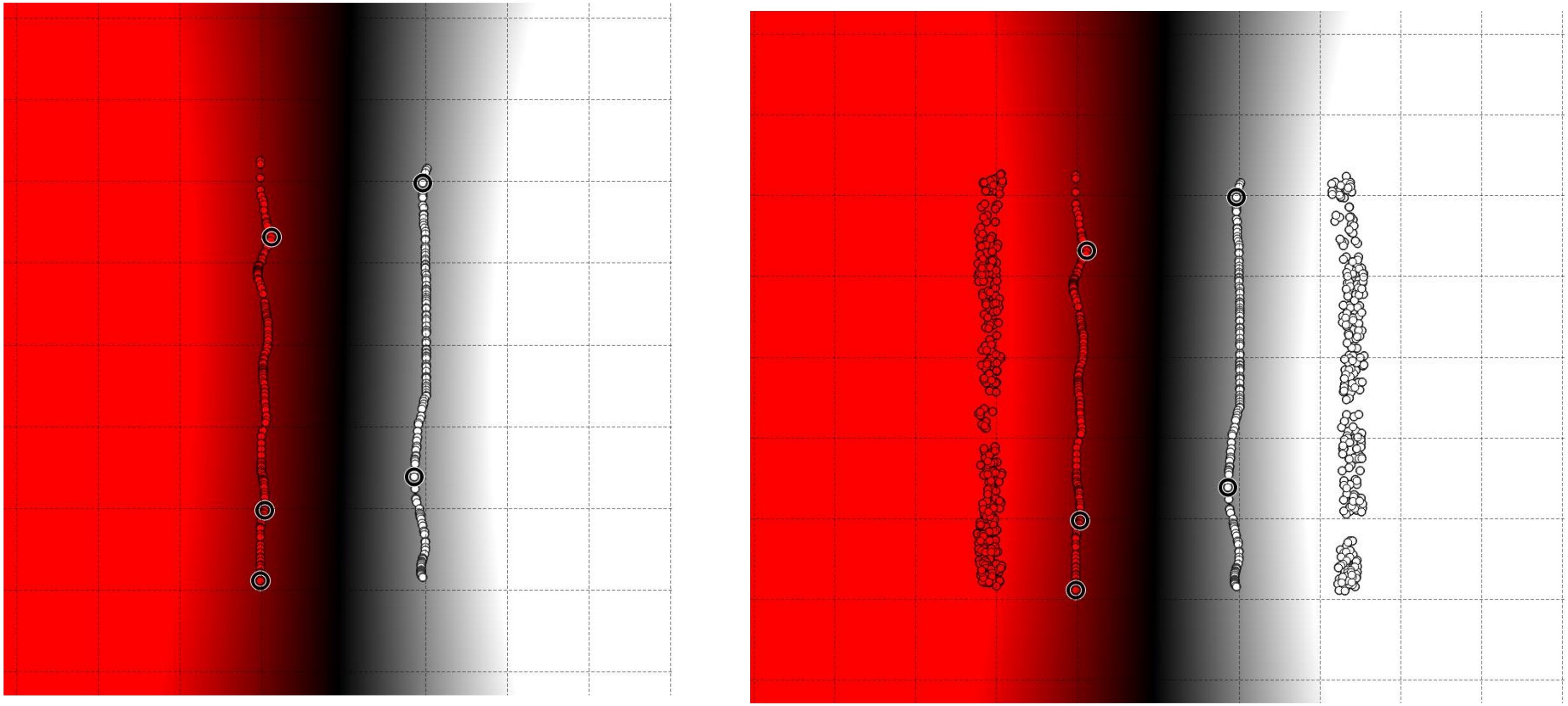
Will the number of Support Vectors (SV) change as we have more datapoints?





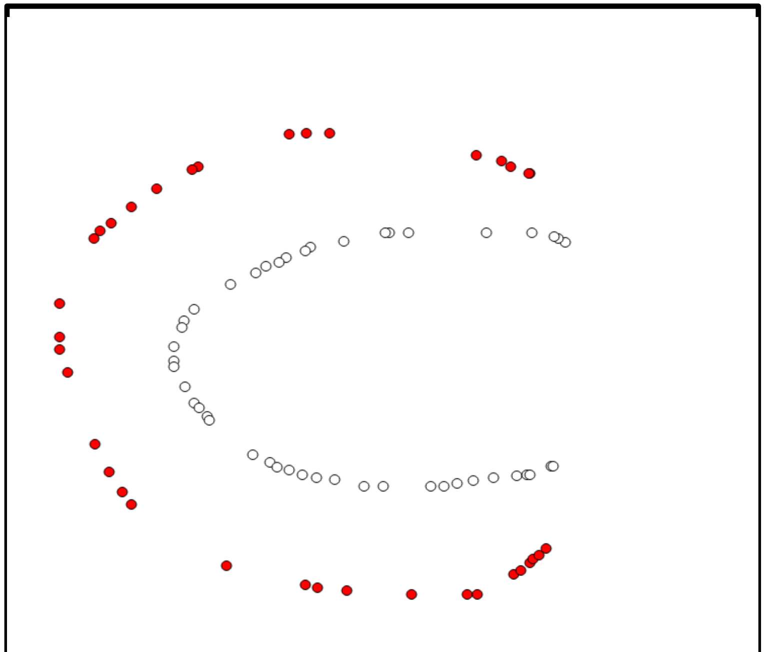
In general, the SVs may be different, but if we have the same number of datapoints close to the border of the margin, the SV-s may be the same, as only the datapoints at the border of the margin matter.

Will the number of Support Vectors (SV) change as we have more datapoints?

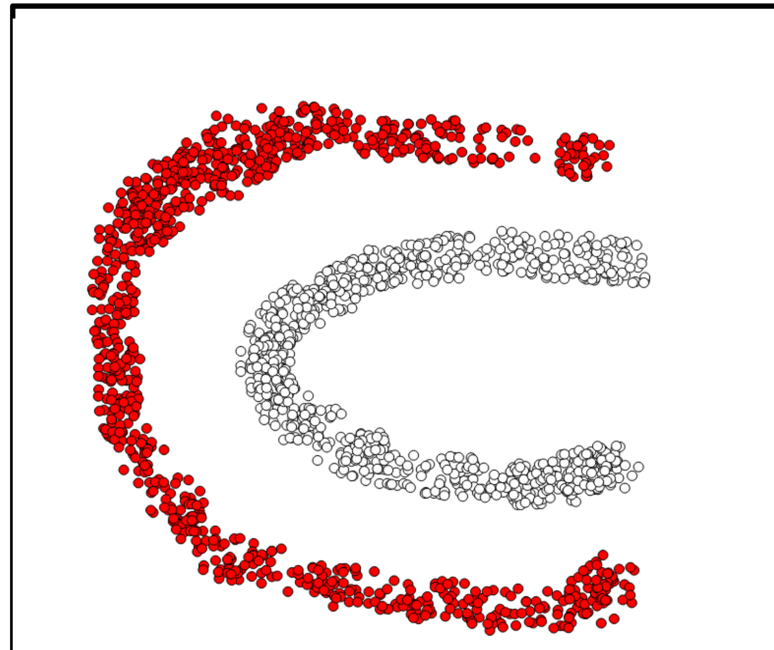


Adding datapoints away from the boundary will not affect classification.

Will the number of Support Vectors (SV) change as we have more datapoints?



(a)



(b)

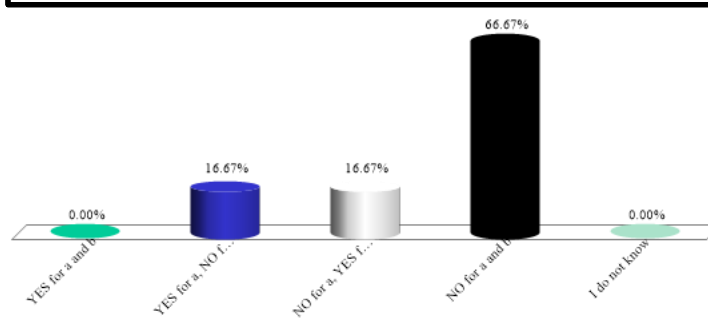
A. YES for a and b

B. YES for a, NO for b

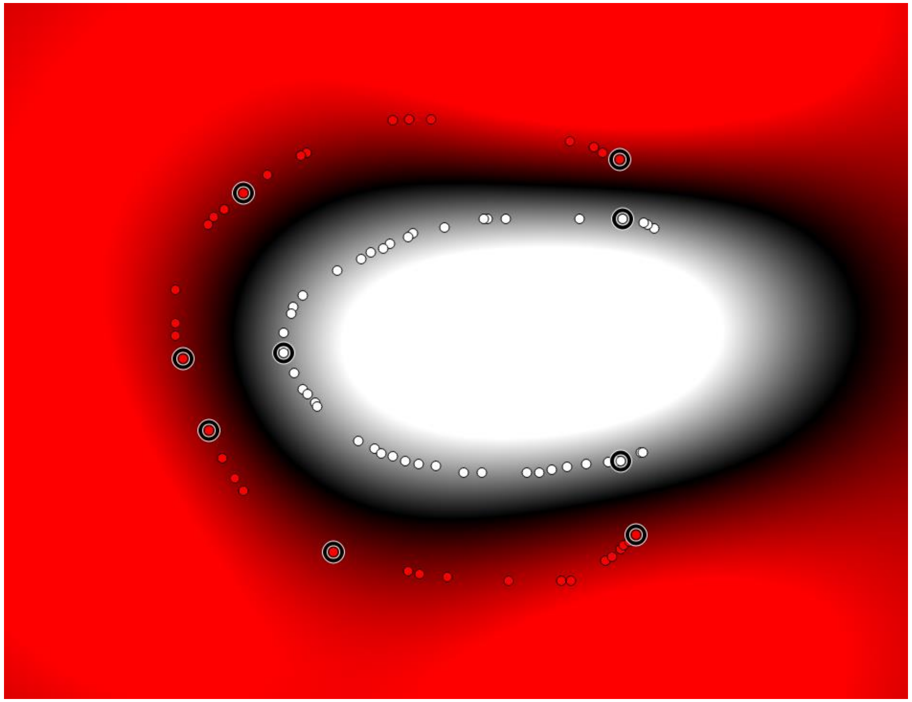
C. NO for a, YES for b

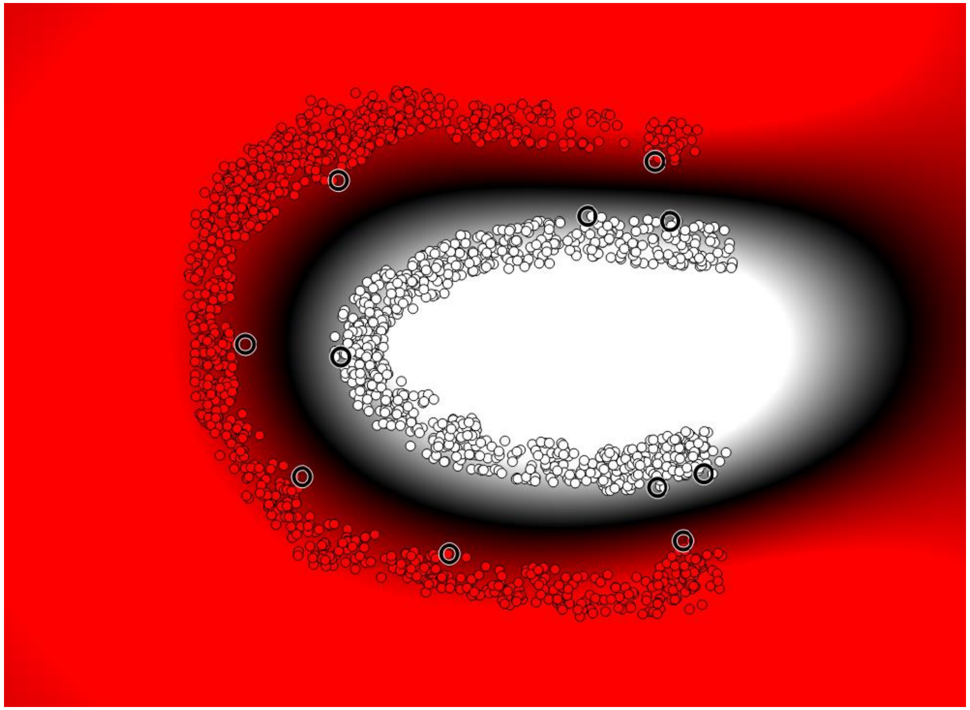
D. NO for a and b

E. I do not know



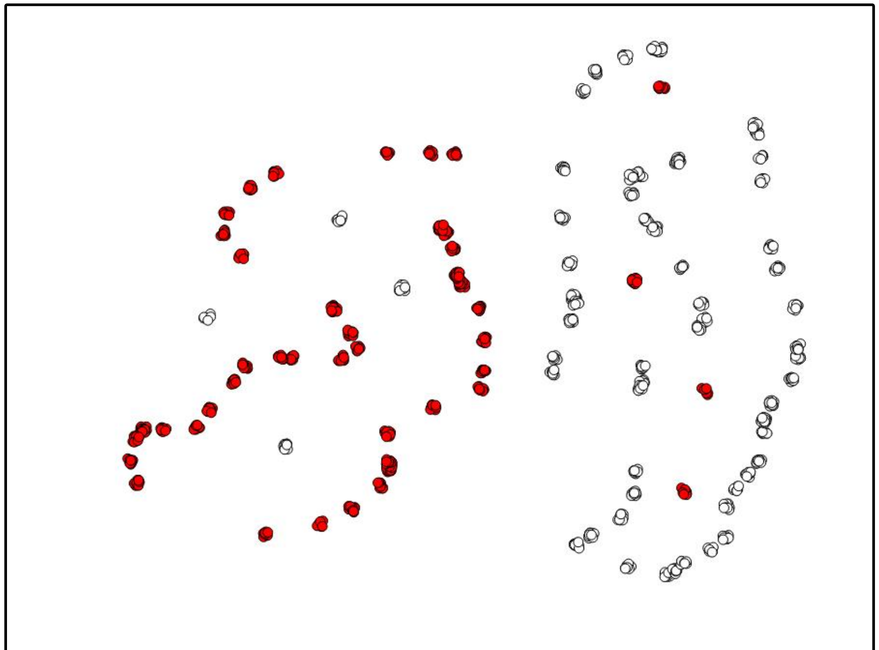
Will the number of Support Vectors (SV) change as we have more datapoints?



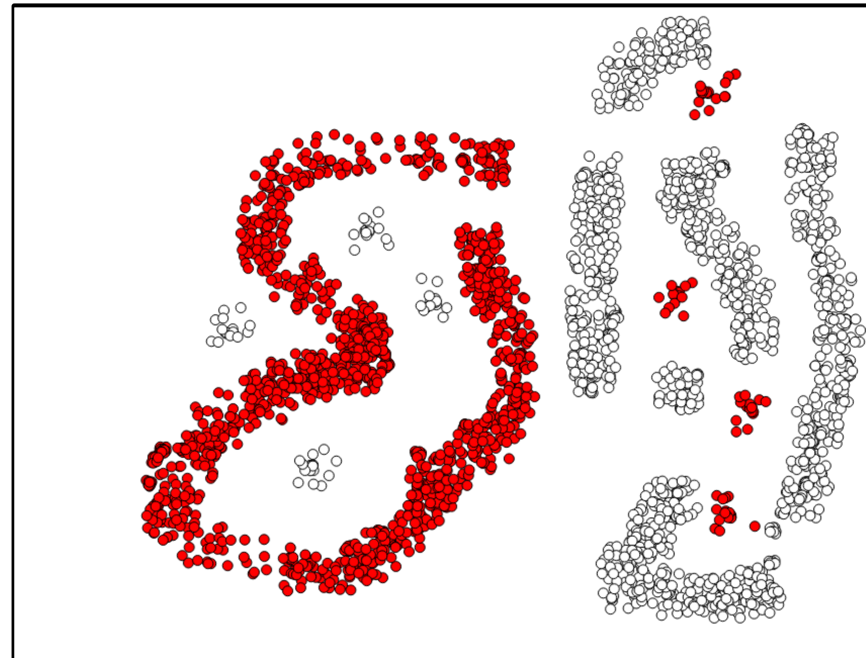


Barely as the boundary is not affected

Will the number of Support Vectors (SV) change as we have more datapoints?

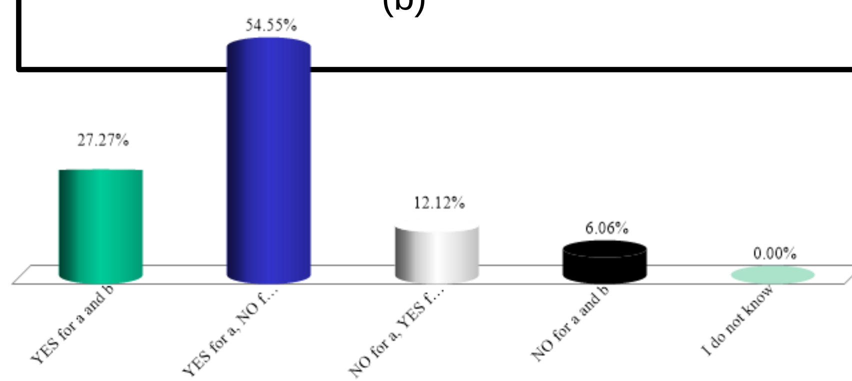


(a)

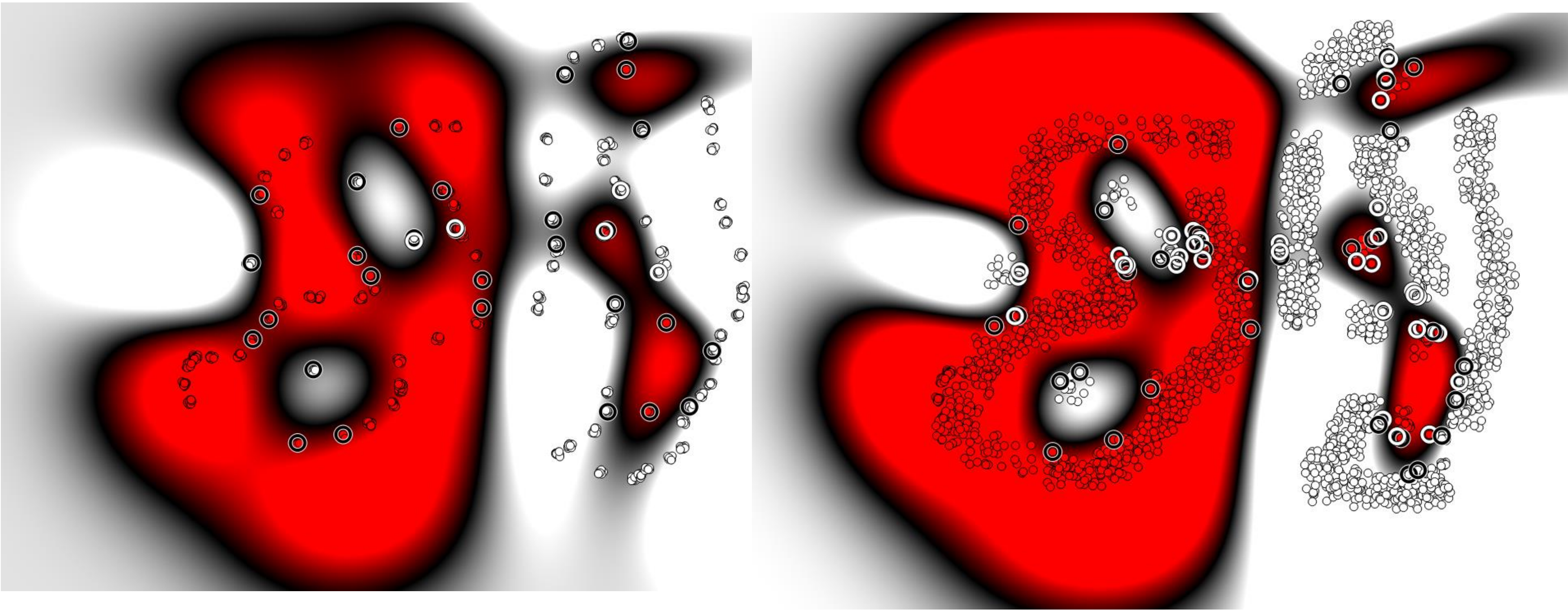


(b)

A. YES for a and b
B. YES for a, NO for b
C. NO for a, YES for b
D. NO for a and b
E. I do not know

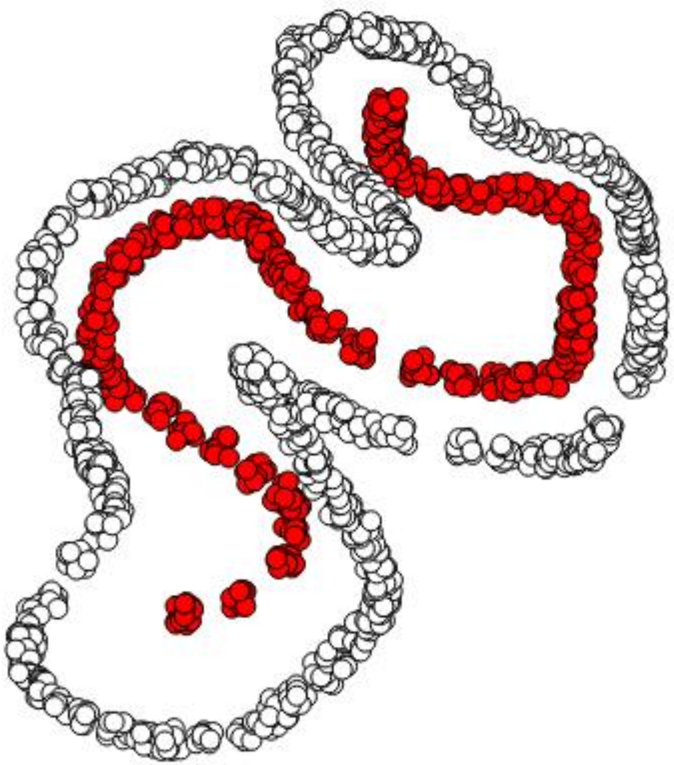


Will the number of Support Vectors (SV) change as we have more datapoints?



Yes, many new SV-s are added. To maintain the boundary around the regions with few datapoints requires a tight fit This implies to use a small kernel width. As a result, each SV has only a local influence.

Non-Linear Classification



The decision function is given by:

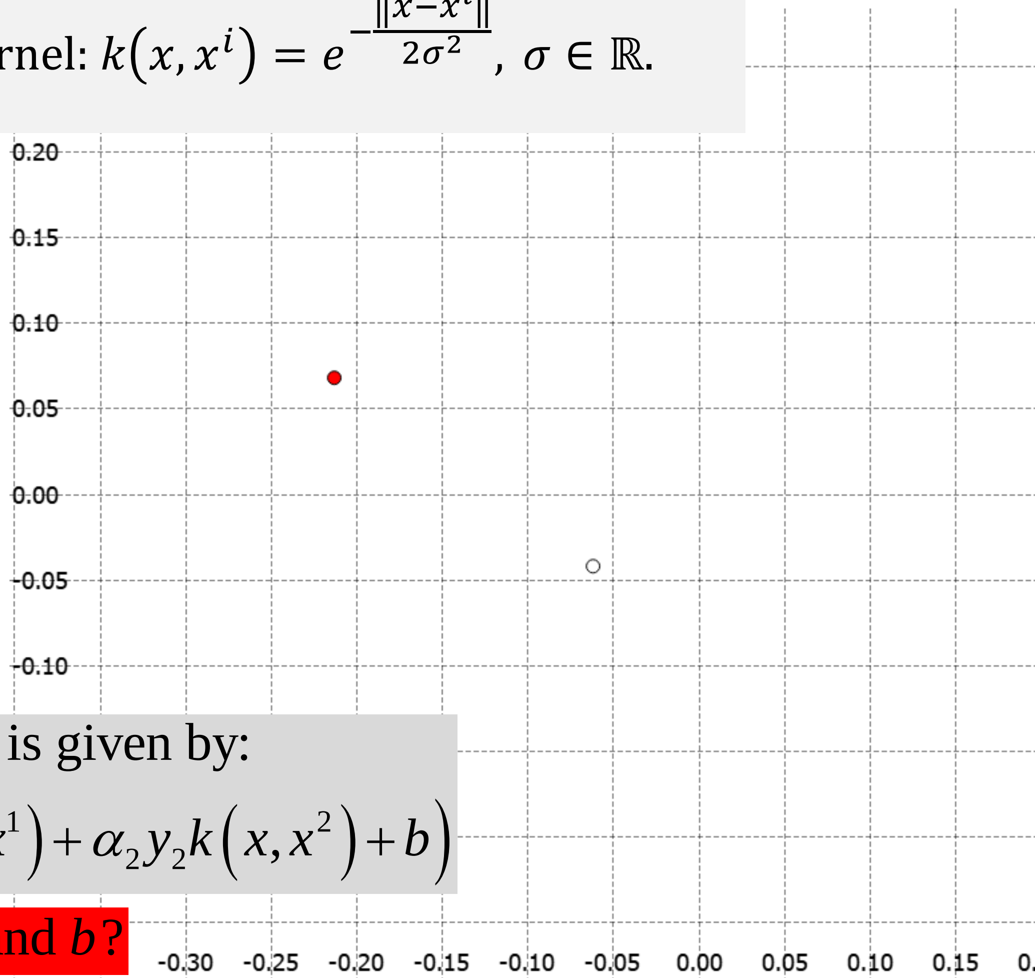
$$f(x) = sgn\left(\sum_{i=1}^{M} \alpha_i y_i k\left(x, x^i\right) + b\right)$$

How can we build the decision boundary?

The decision function is given by:

$$f(x) = sgn\left(\sum_{i=1}^{M} \alpha_i y_i k\left(x, x^i\right) + b\right)$$

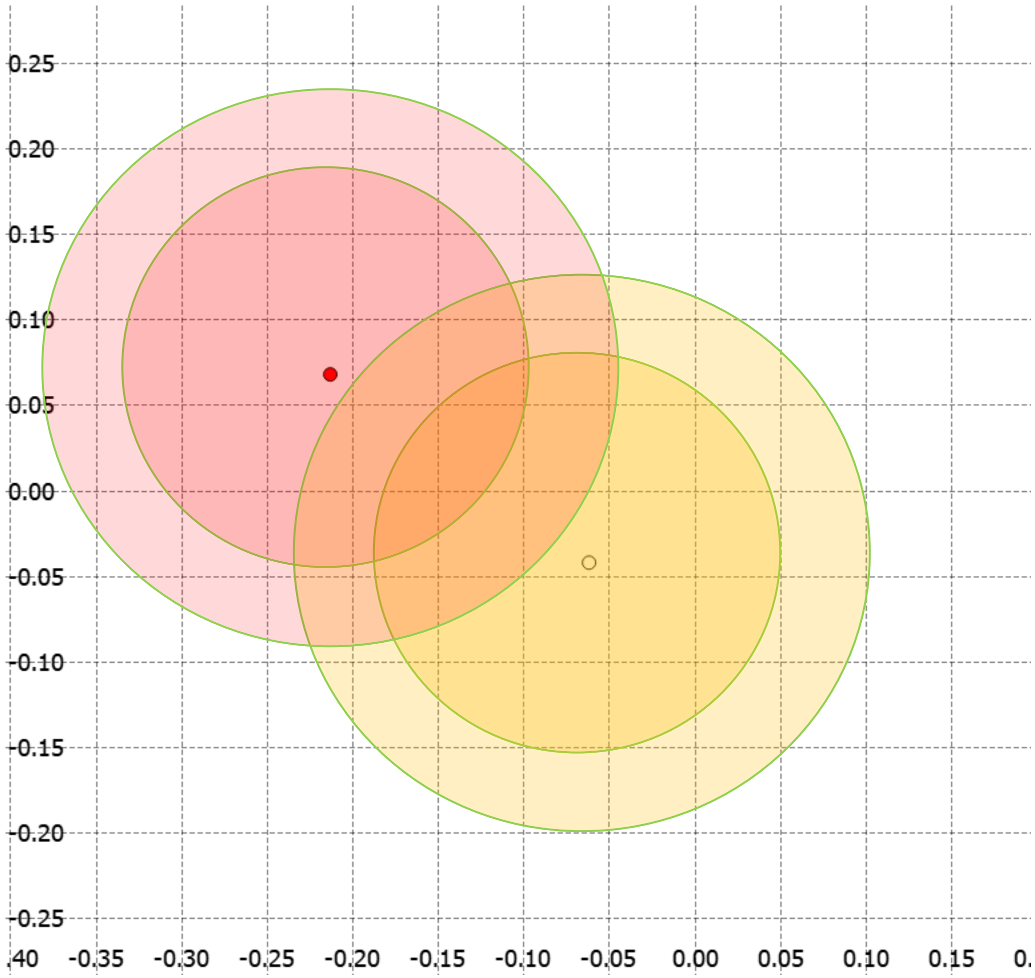
RBF (Gaussian) kernel: $k\left(x, x^i\right) = e^{-\frac{\left\|x - x^i\right\|}{2\sigma^2}}$, $\sigma \in \mathbb{R}$.



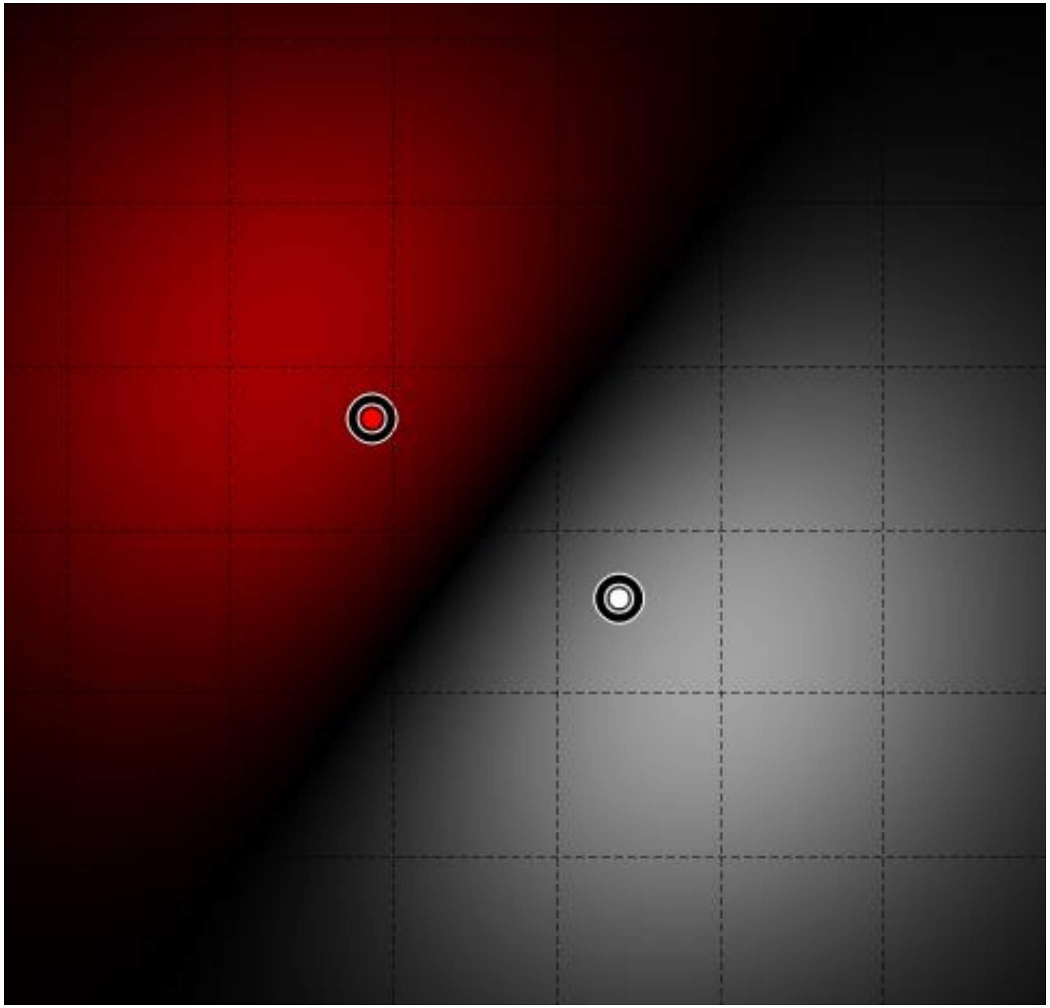
The decision function is given by:

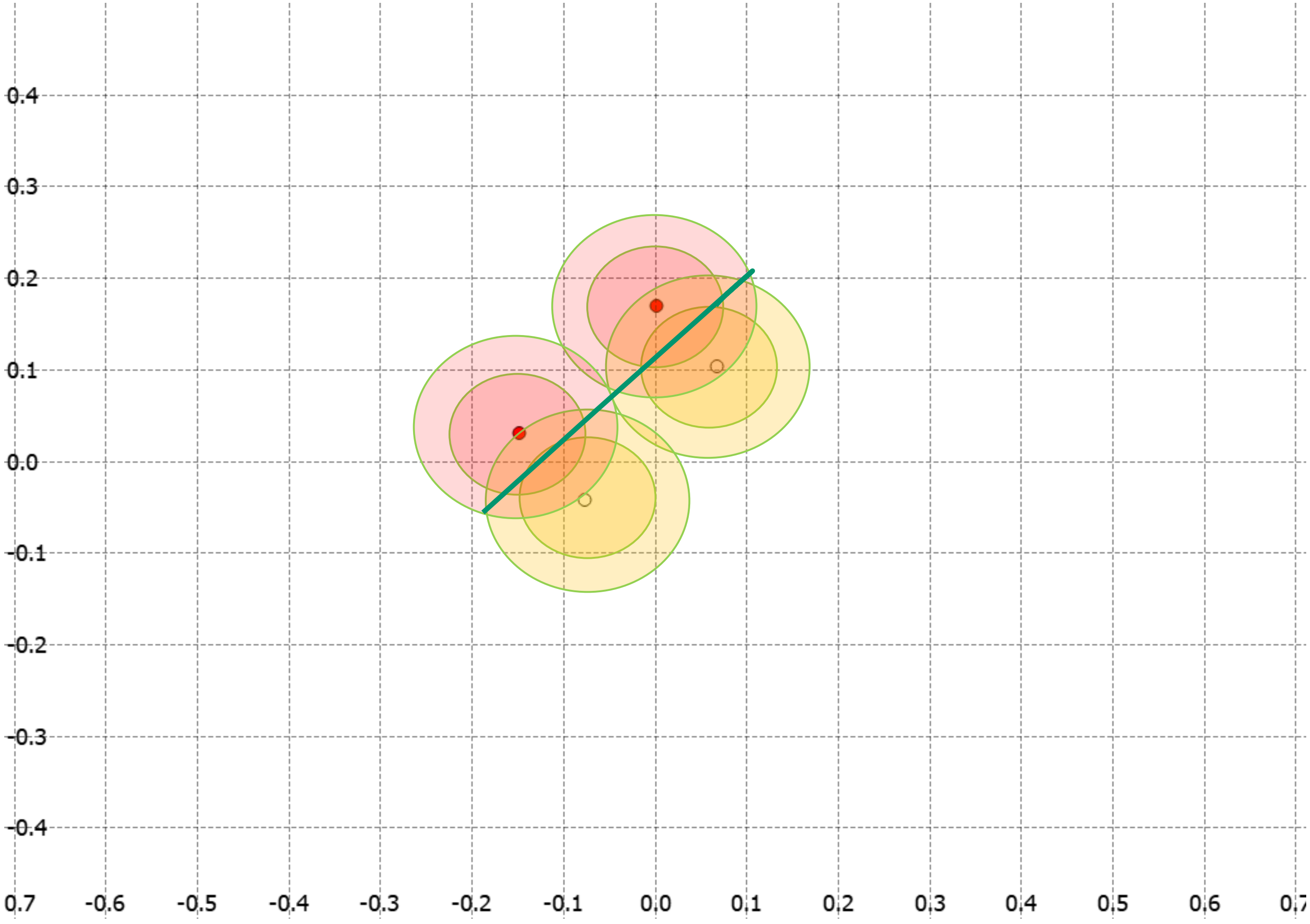
$$f(x) = sgn\left(\alpha_1 y_1 k\left(x, x^1\right) + \alpha_2 y_2 k\left(x, x^2\right) + b\right)$$

What are α_1, α_2 and b?



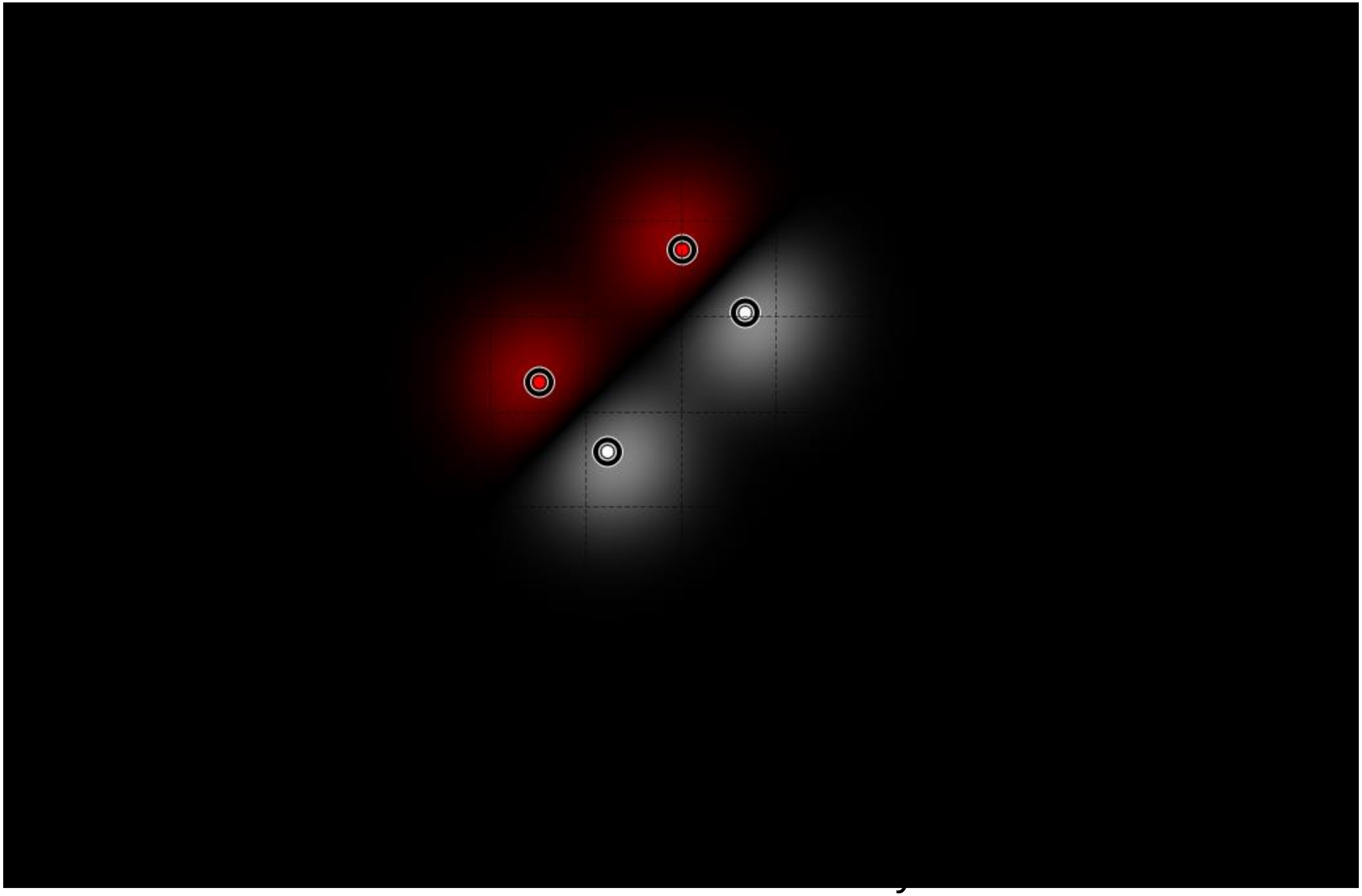
Draw the decision boundary assuming equal effect of both points

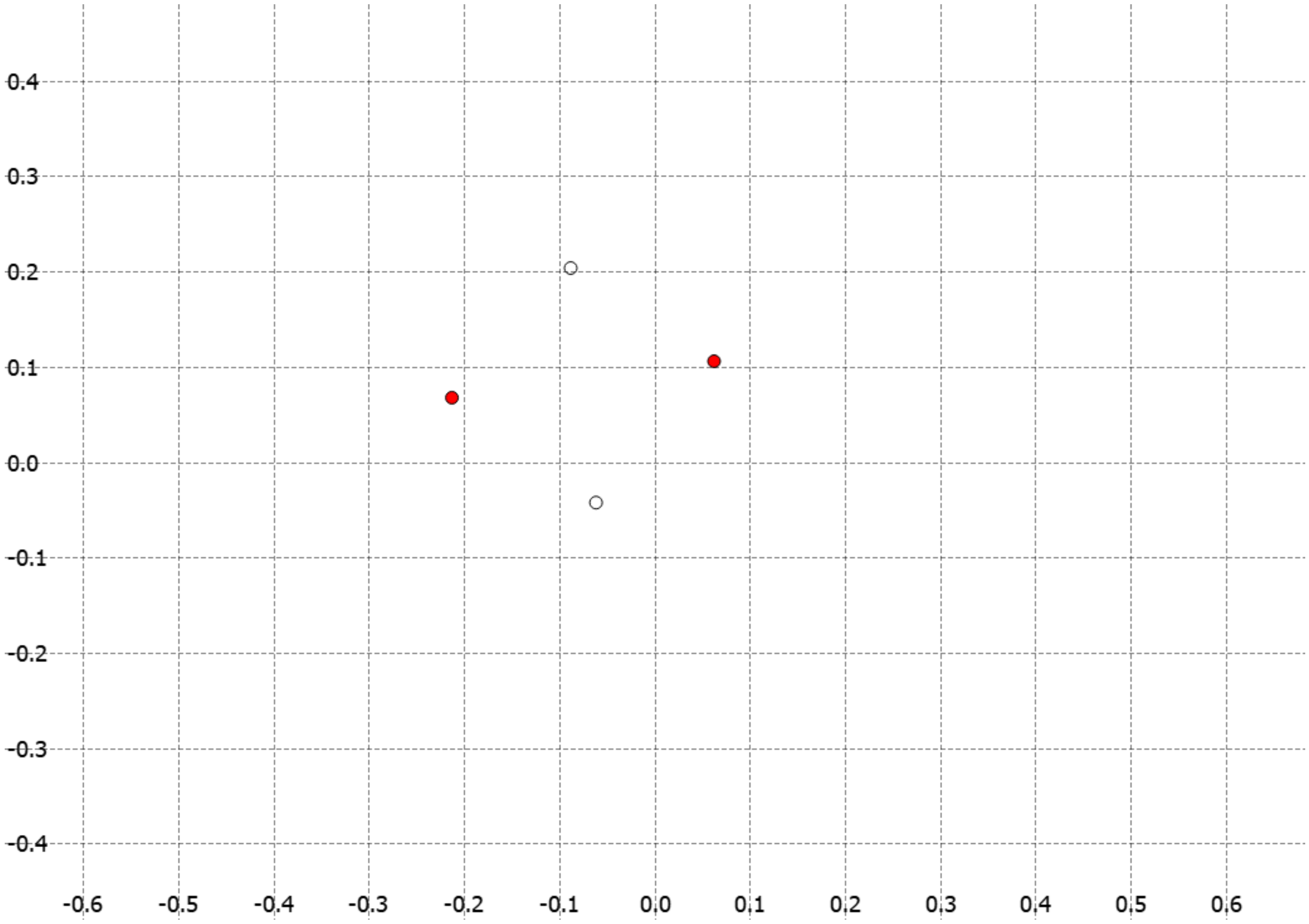




Draw the decision boundary?

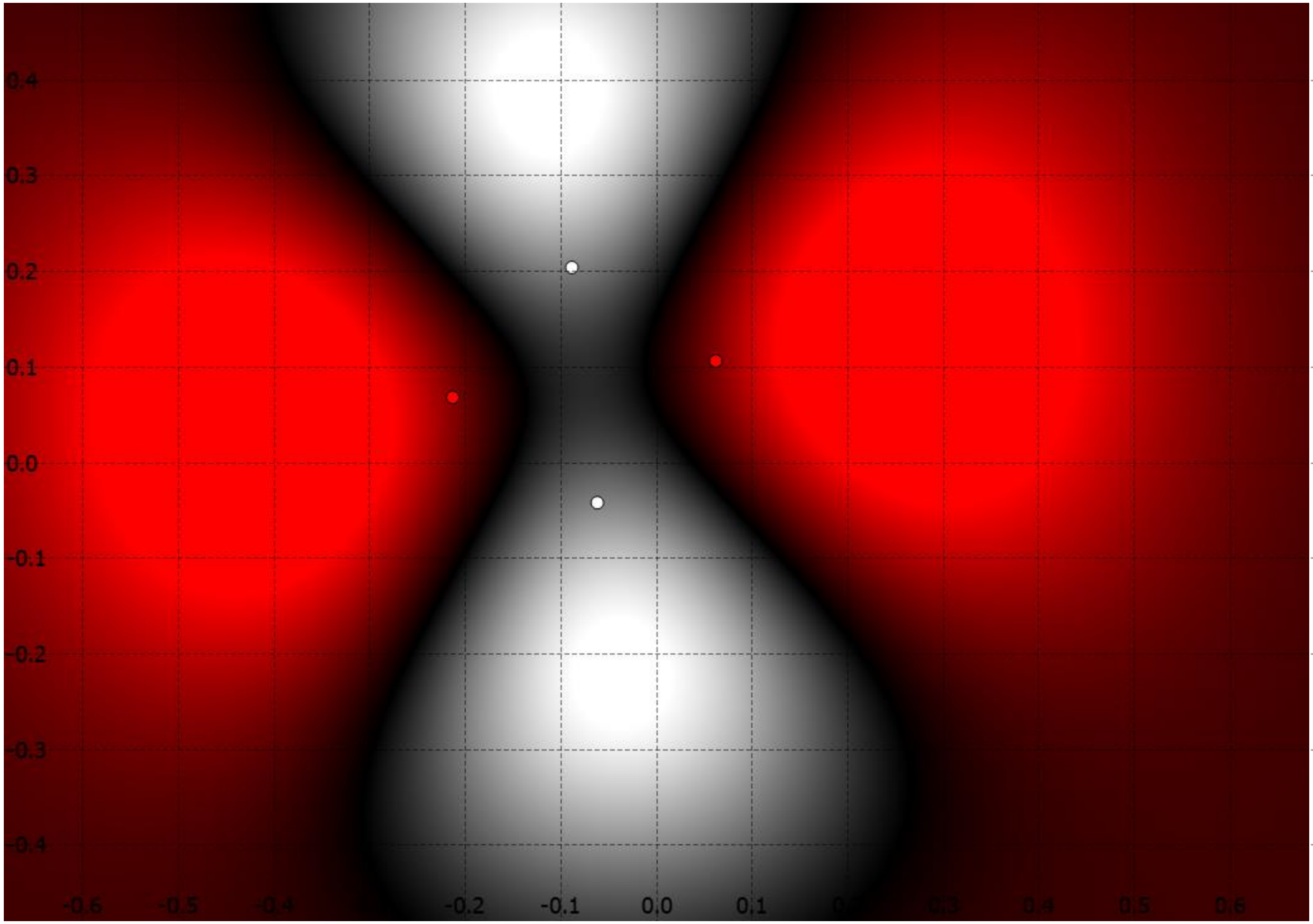
Solution





Draw the decision boundary?

Solution

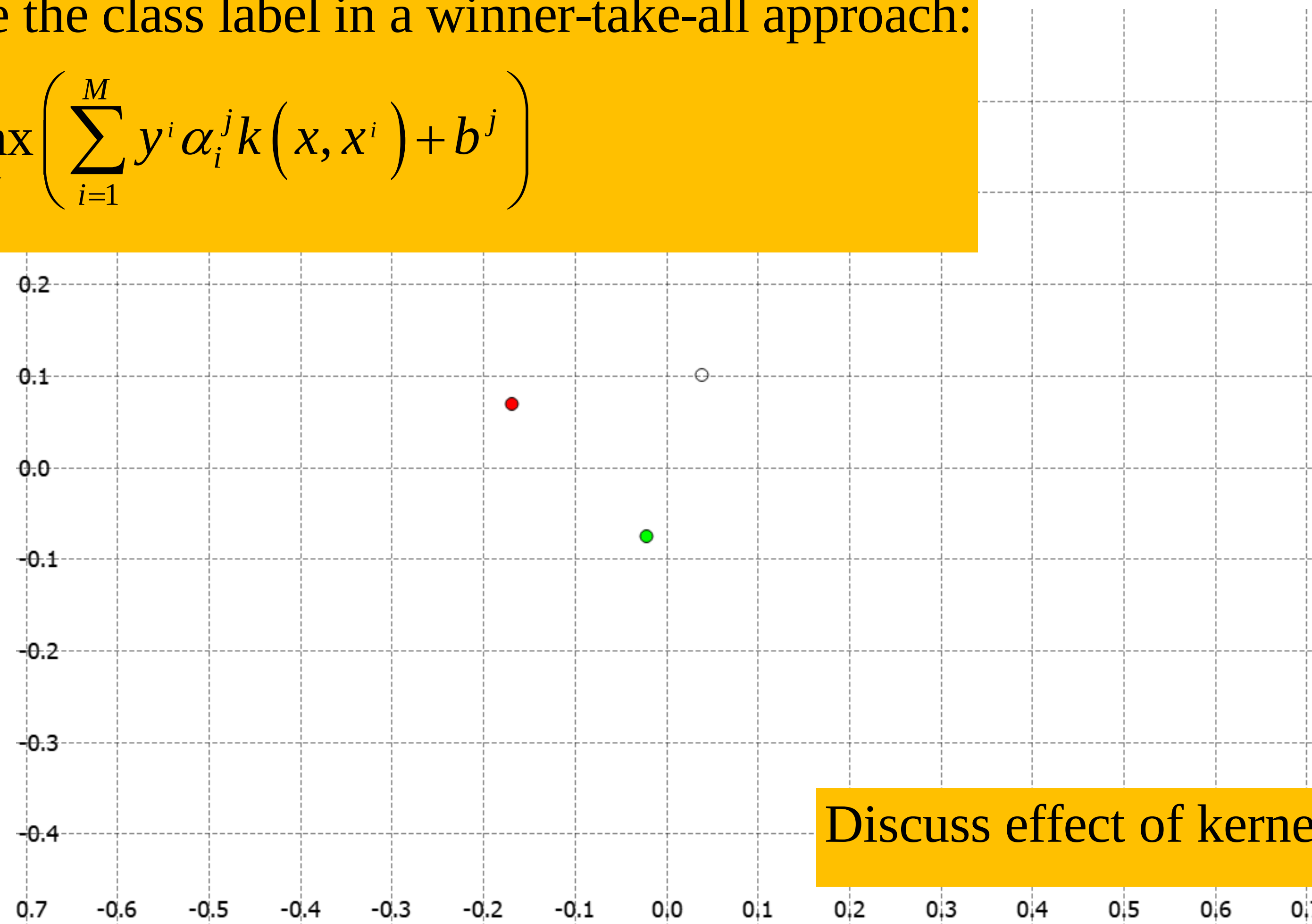


Draw the decision boundary?

Multi-Class SVM

Compute the class label in a winner-take-all approach:

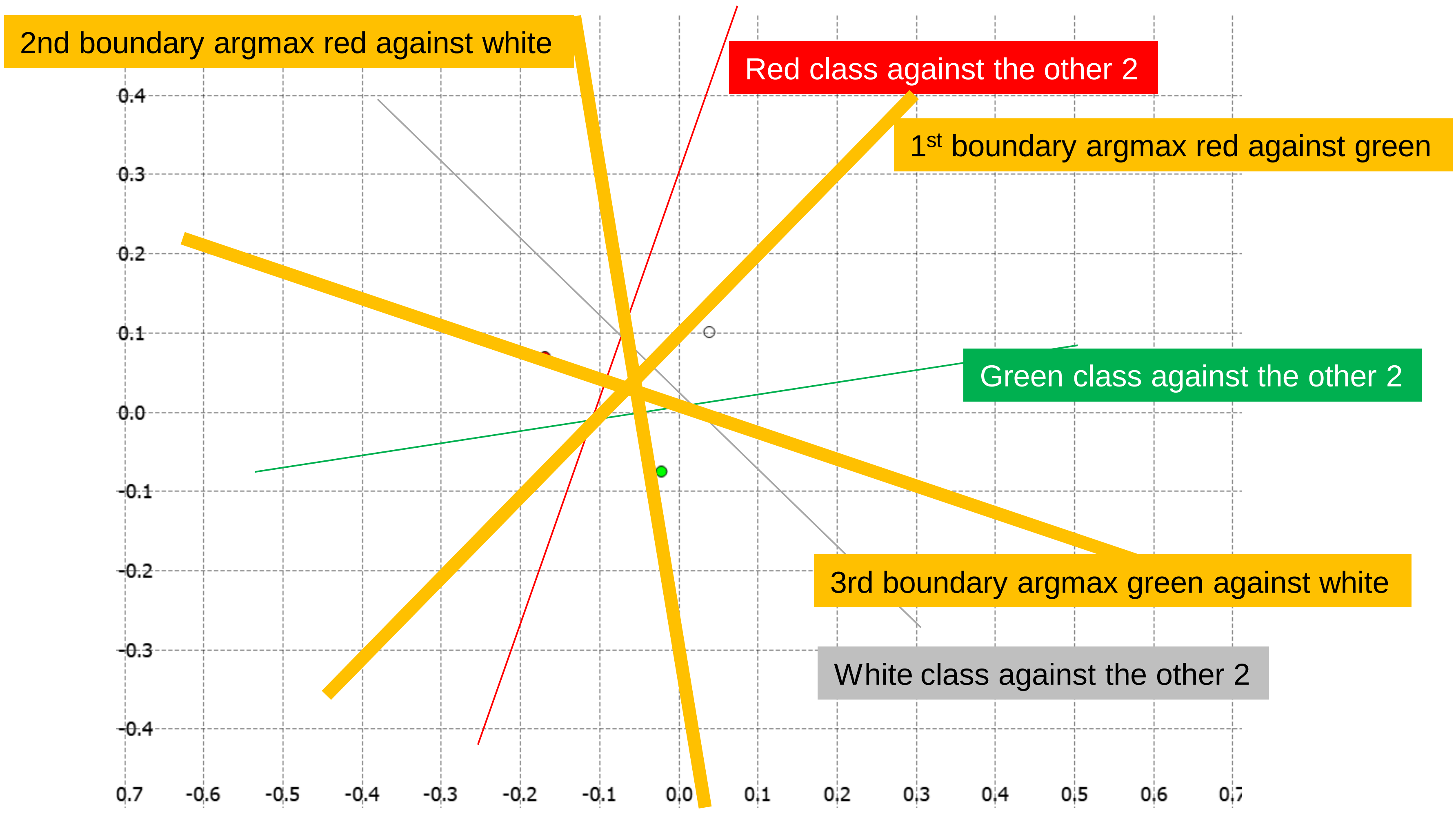
$$j = \arg\max_{j=1,\dots K} \left(\sum_{i=1}^{M} y^i \alpha_i^j k\left(x, x^i\right) + b^j \right)$$



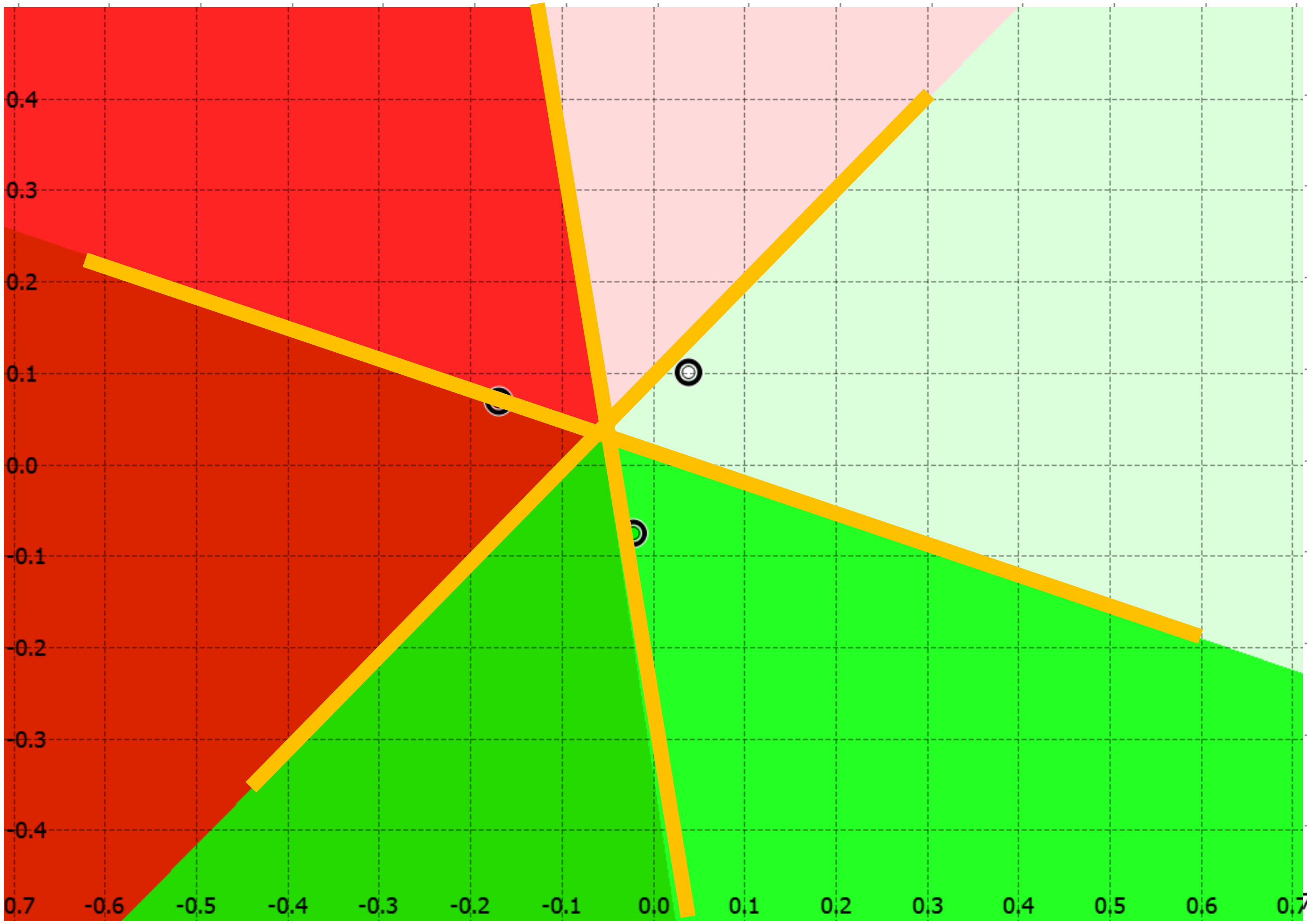
Discuss effect of kernel width σ

Draw the decision boundary (assume RBF kernel)?

Multi-Class SVM

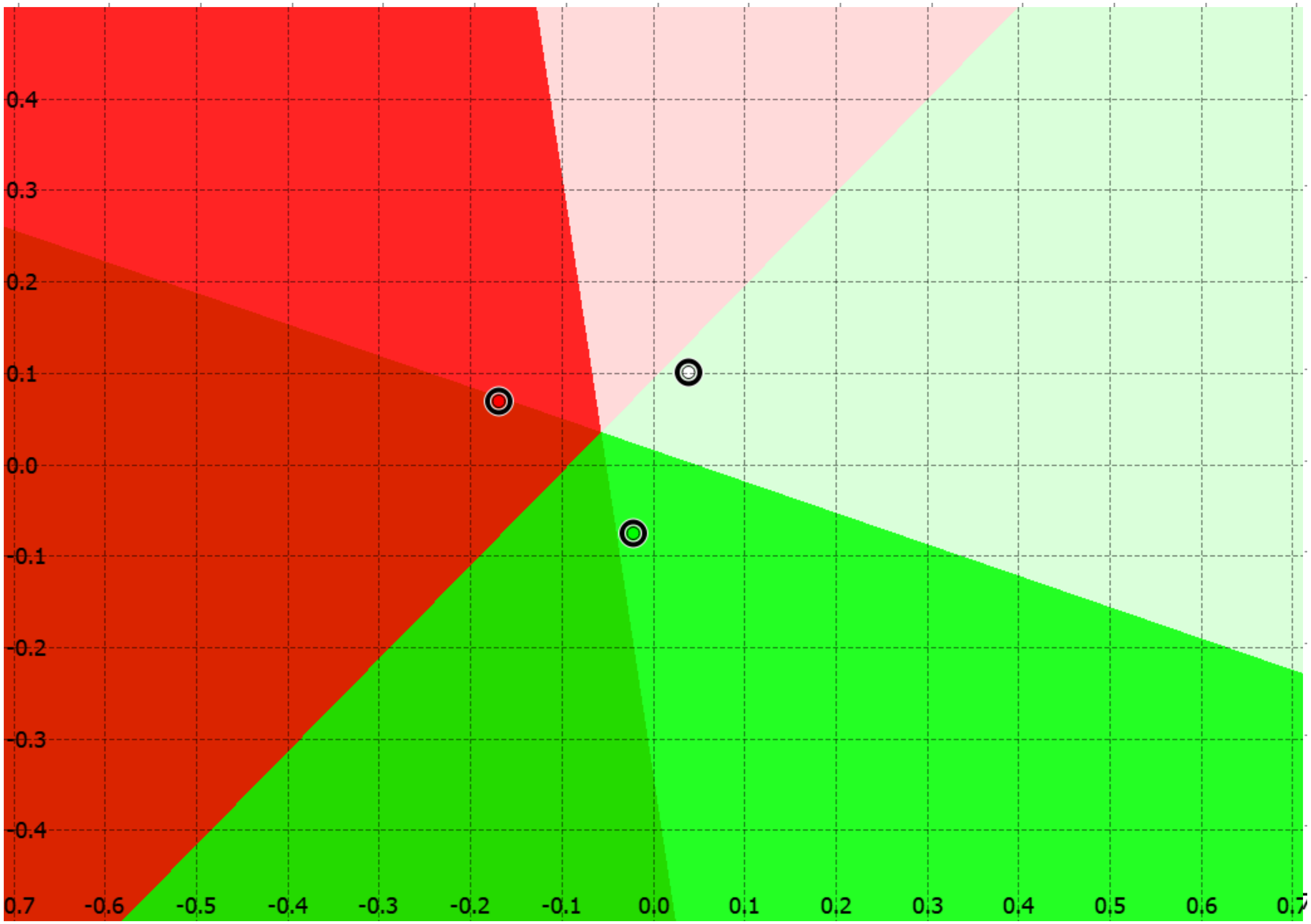


Multi-Class SVM



Solution ~ equivalent to linear SVM

Multi-Class SVM



Solution ~ equivalent to linear SVM

Multi-Class SVM

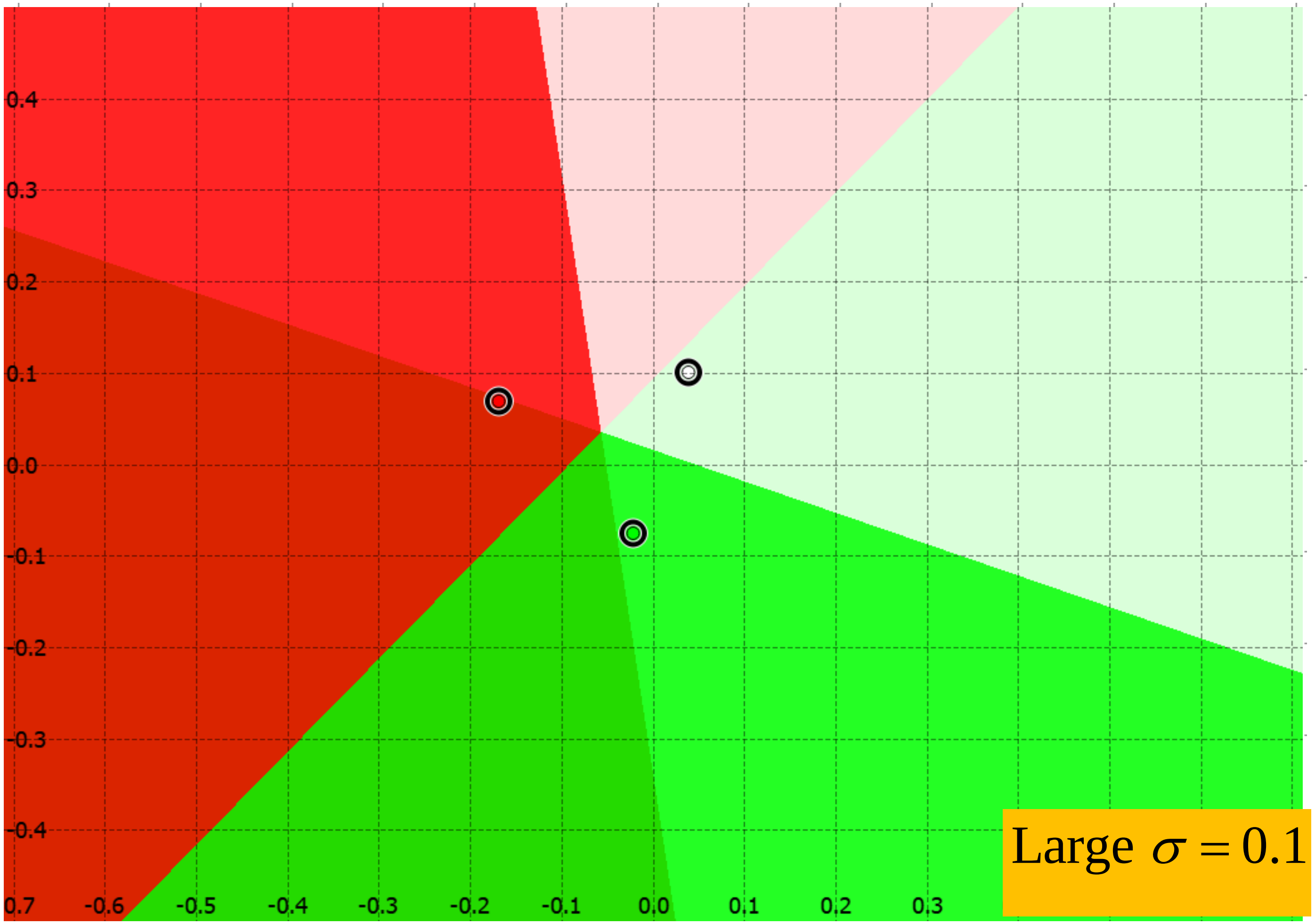
Discuss effect of kernel width σ

Compute the class label in a winner-take-all approach:

$$j = \arg\max_{j=1,\ldots K} \left(\sum_{i=1}^{M} y^i \alpha_i^j \underbrace{k\left(x, x^i\right)}_{\sim 0 \text{ when } \left\|x - x^i\right\| \to \infty} + b^j \right) = \arg\max_{j=1,\ldots K} \left(b^j\right)$$

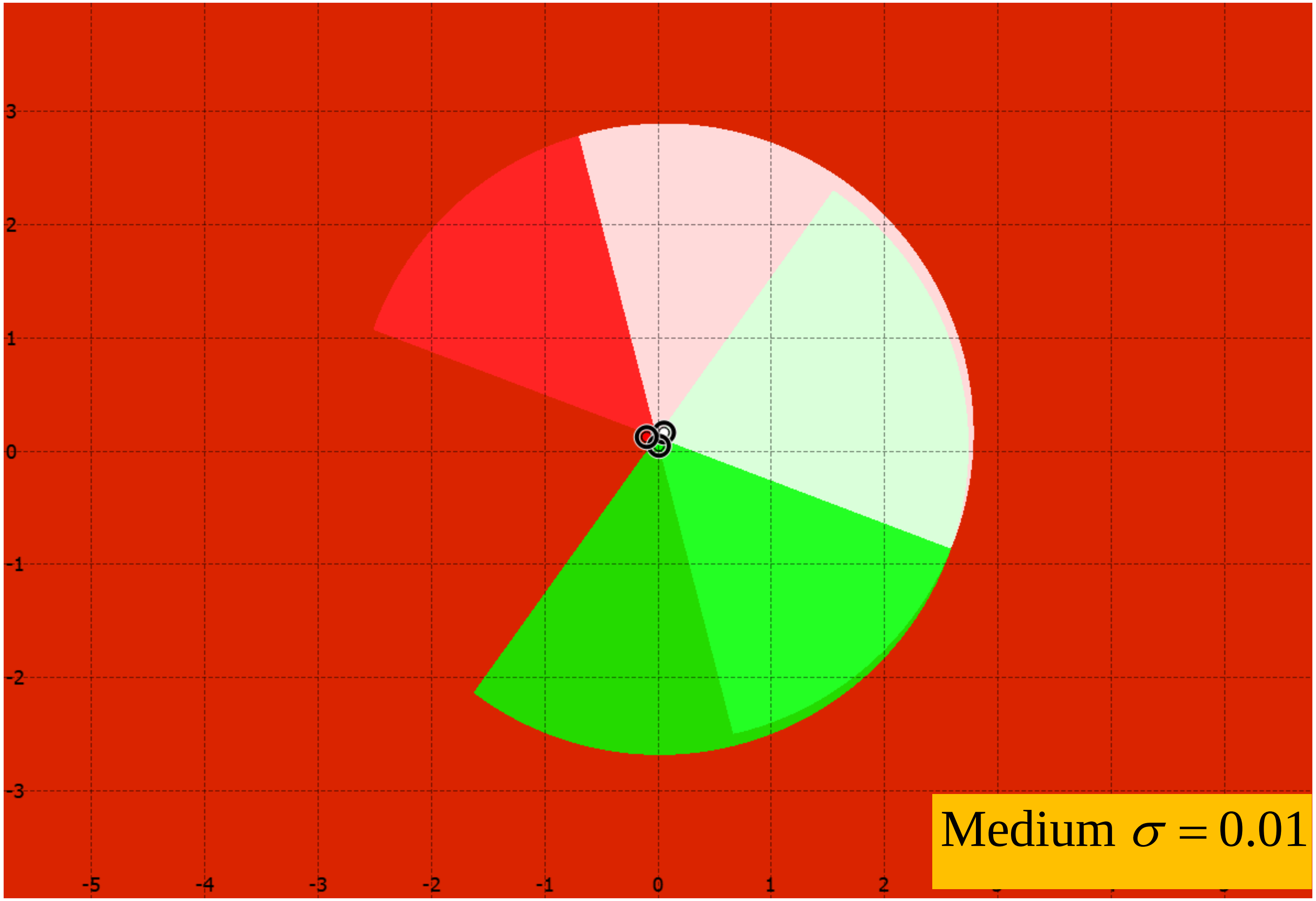
The kernel width does not affect this boundary except when very far from the datapoints where numerically the RBF function becomes zero and prediction is based on b

Multi-Class SVM



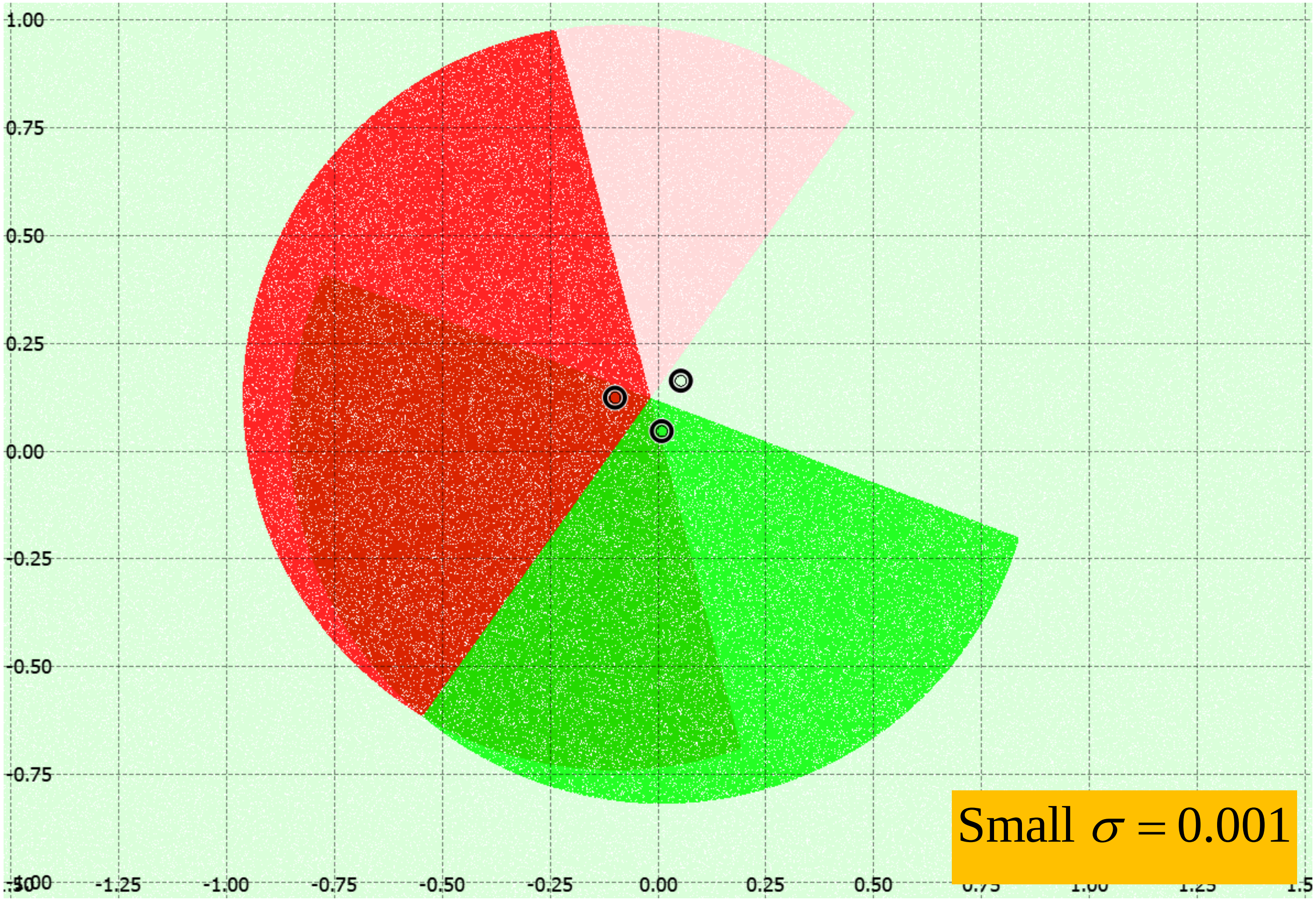
Solution ~ equivalent to linear SVM

Multi-Class SVM



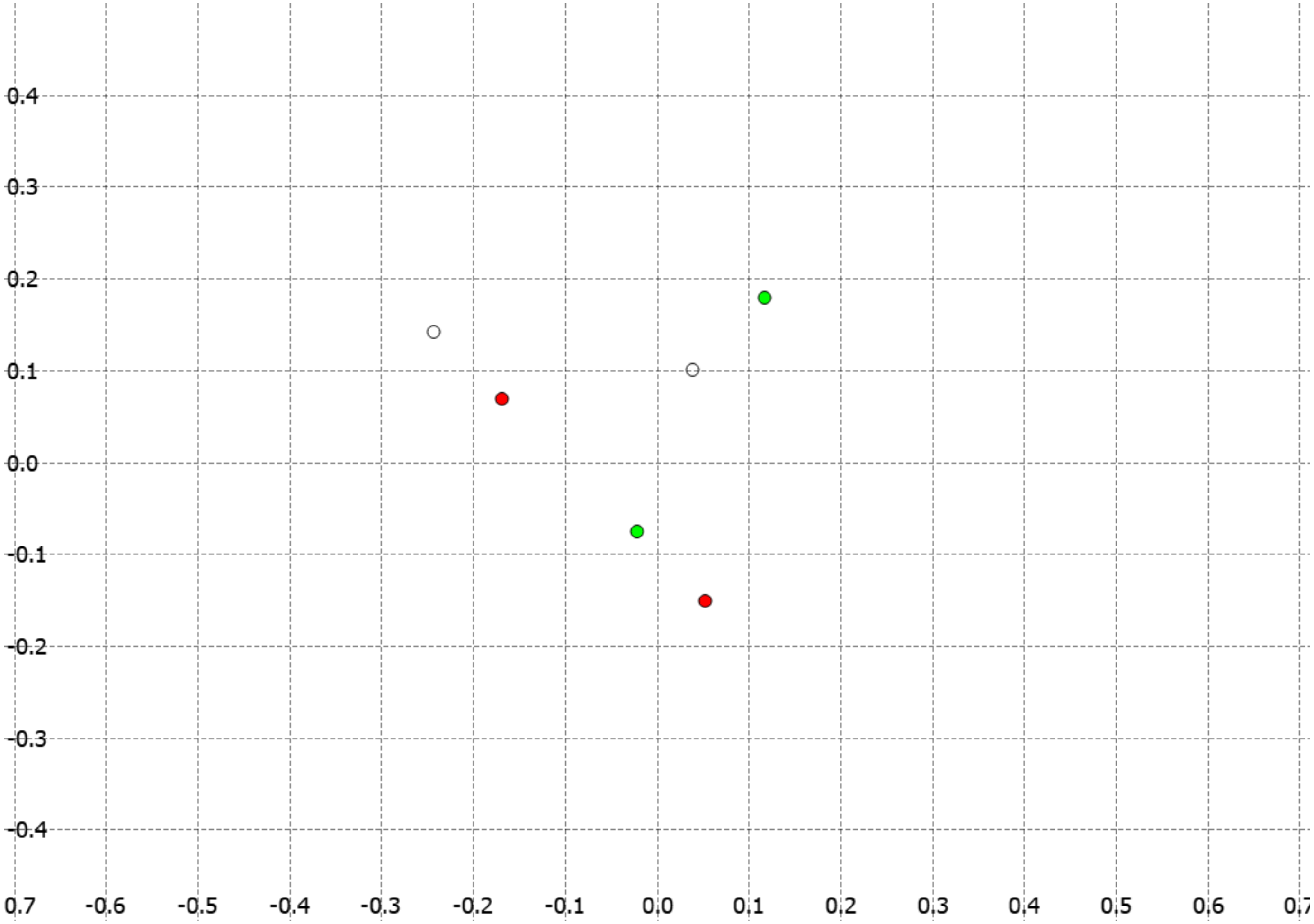
Far from datapoints, the red class labels wins.

Multi-Class SVM



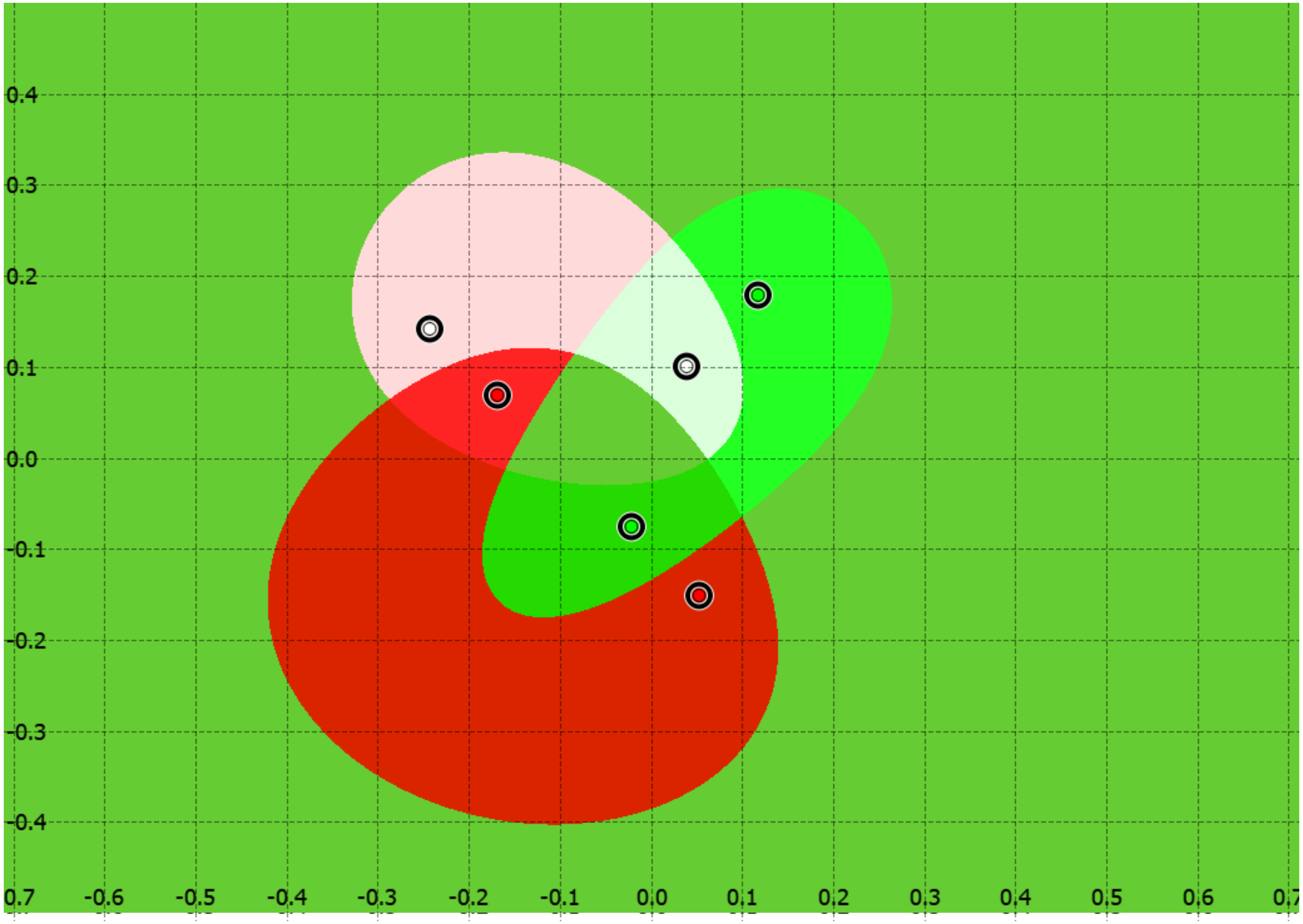
The smaller the kernel width, the earlier in space the wrong label appears, as the RBF vanishes faster.

Multi-Class SVM



Draw the decision boundary (assume RBF kernel)?

Multi-Class SVM



Solution