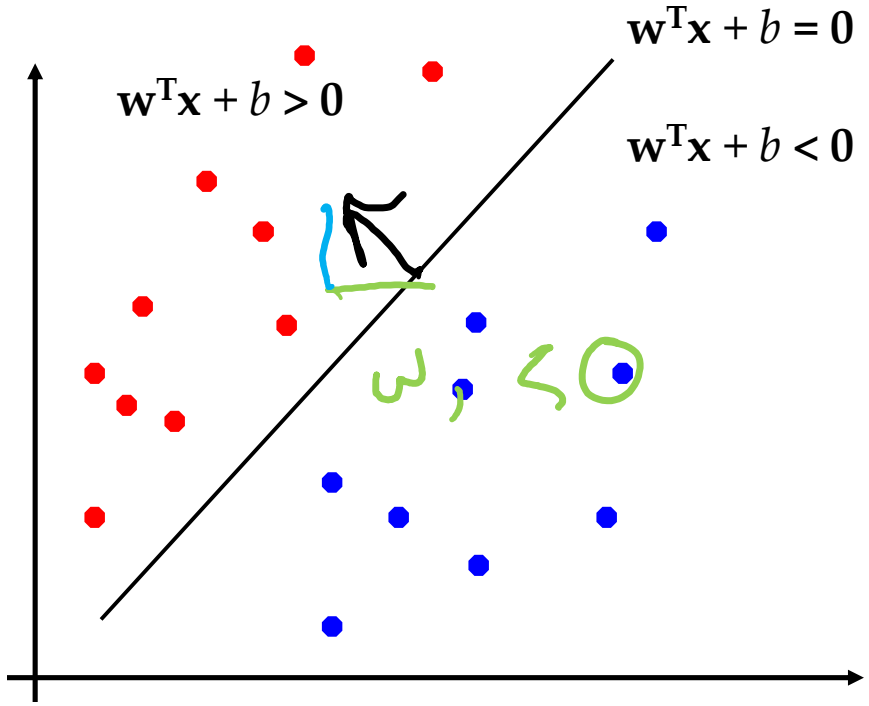
CS 412

FEB 18TH –SVM + NEURAL NETWORKS

HTF – CHAPTER 12 + CHAPTER 11

Linear Separators

Binary classification can be viewed as the task of separating classes in feature space:



$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + b)$

Hyperplane that correctly separates

$$\mathcal{X} = \left\{\mathbf{x}^t, r^t\right\}_t \text{ where } r^t = \begin{cases} +1 & \text{if } \mathbf{x}^t \in C_1 \\ -1 & \text{if } \mathbf{x}^t \in C_2 \end{cases}$$

find $\mathbf{w}$ and w_0 such that

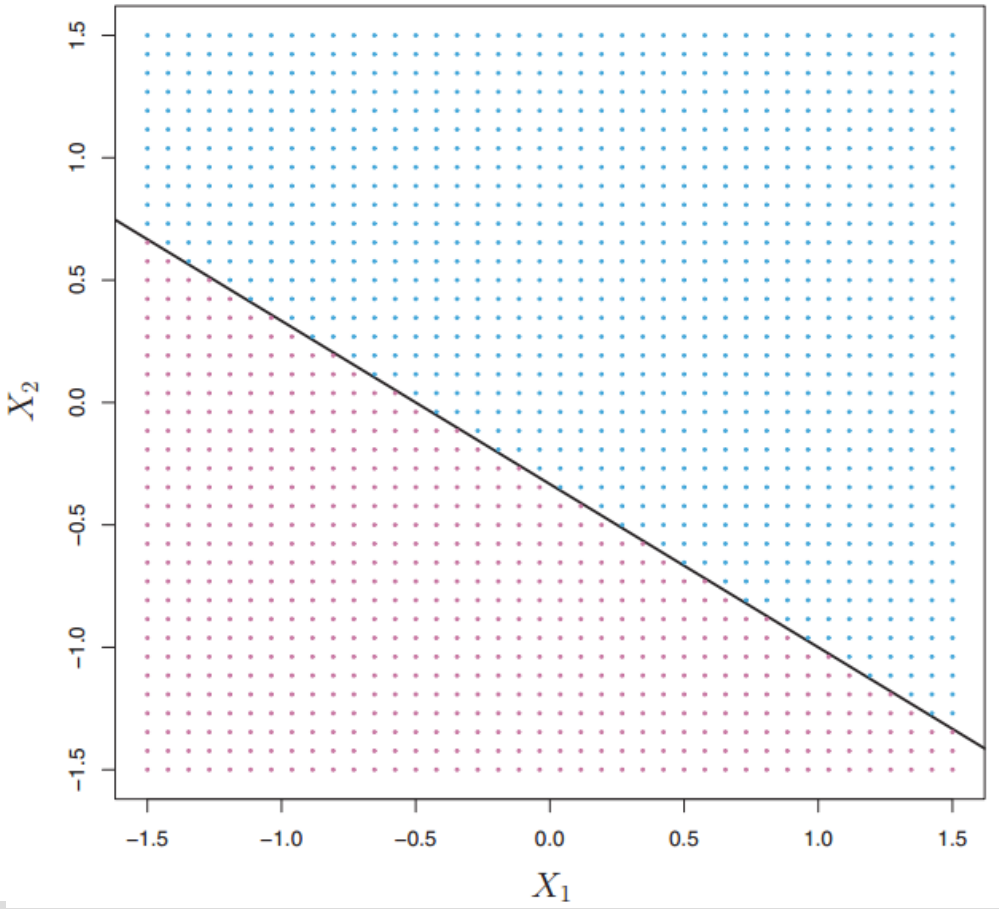
$\mathbf{w}^T\mathbf{x}^t + w_0 \geq 0$ for $r^t = +1$

$\mathbf{w}^T\mathbf{x}^t + w_0 \leq 0$ for $r^t = -1$

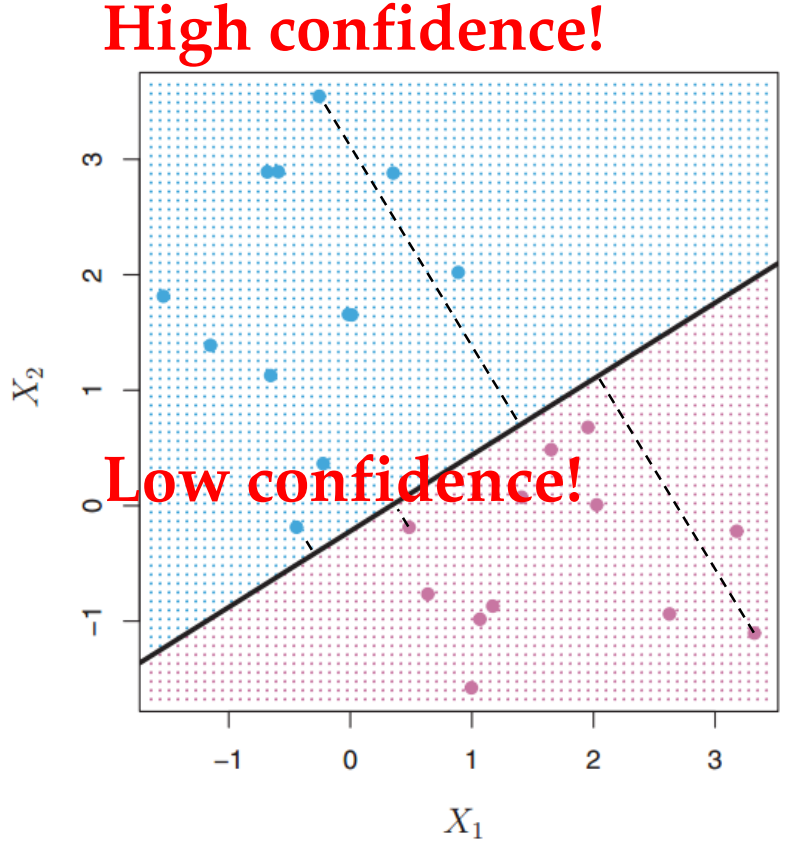
which can be rewritten as

$r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right) \geq +1$

- Usually no solutions (not linearly separable)
- But…assume there is a solution, then what?



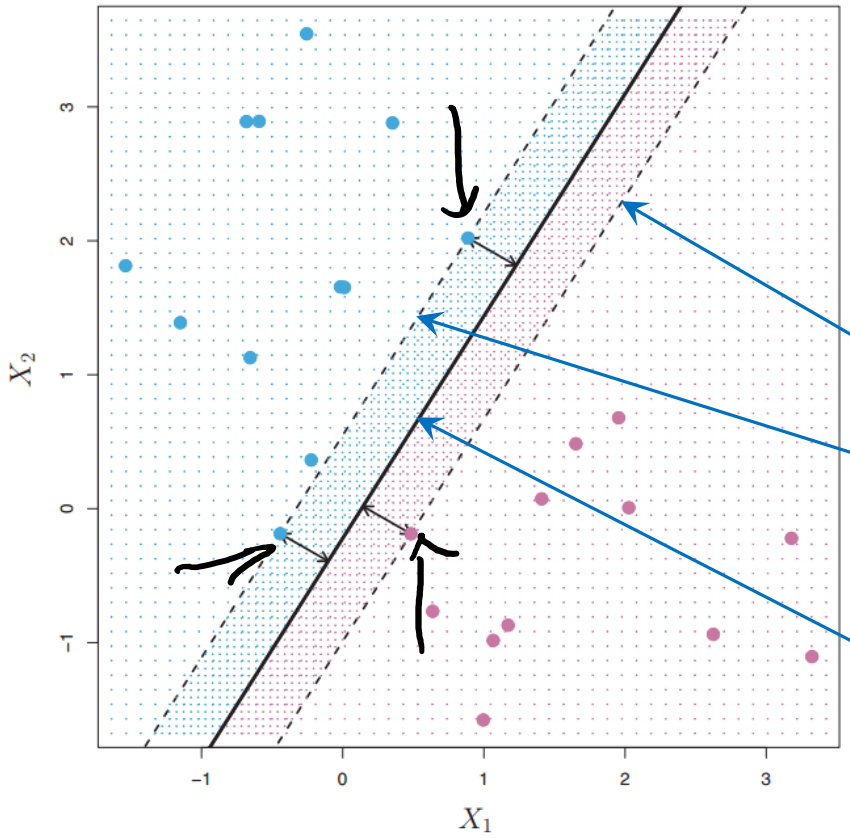
"Confidence" of Predictions



"Confidence" = $r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right)$

What about multiplying $\mathbf{w}$ and w_0 by 2 or 100?

Pick the one with the largest margin!



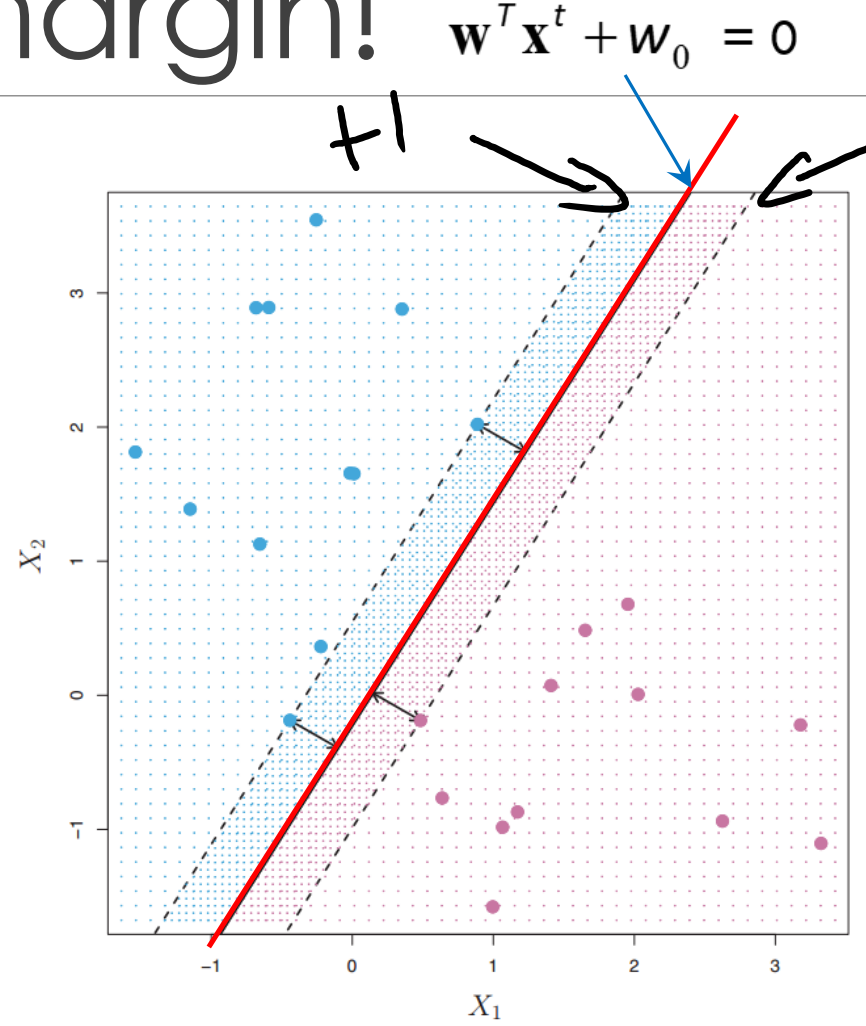
Points on the margin boundary have the lowest "confidence" over all points

Let's maximize this!

margin boundaries

$\mathbf{w}^T\mathbf{x}^t + w_0 = 0$ separation boundary

Pick the one with the largest margin!



Points on the margin boundary have the lowest "confidence" over all points

Let's maximize this!

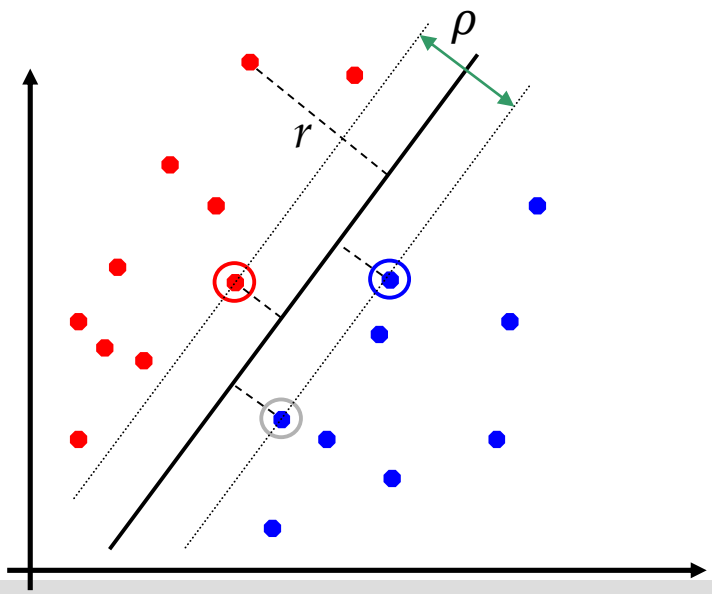
Naturally, we want the margin to be the same for pos and neg

Classification Margin

Distance from example $\mathbf{x}_i$ to the separator is

Examples closest to the hyperplane are ***support vectors***.

Margin ρ of the separator is the distance between support vectors.



$$r = \frac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$$

Linear SVMs Mathematically

Then we can formulate the *quadratic optimization problem:*

Find $\mathbf{w}$ such that

$\rho = \frac{2}{\|\mathbf{w}\|}$ is maximized

and for all $(\mathbf{x}_i, y_i)$, i=1..n : $y_i(\mathbf{w}^T\mathbf{x}_i) \geq 1$

Which can be reformulated as:

Find $\mathbf{w}$ such that

$\Phi(\mathbf{w}) = ||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w}$ is minimized

and for all $(\mathbf{x}_i, y_i)$, i=1..n : $y_i(\mathbf{w}^T\mathbf{x}_i) \geq 1$

hard margin
no points are
in the margins

Hard margin SVM (linearly separable)

- Distance from the discriminant to the closest instances on either side
- Distance of x to the hyperplane is $\frac{\left|\mathbf{w}^T\mathbf{x}^t + w_0\right|}{\|\mathbf{w}\|}$
- We require $\frac{r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right)}{\|\mathbf{w}\|} \geq \rho, \forall t$.
  - ρ: margin of the dataset (invariant to scaling of w)
- For a unique sol'n, fix ρ $||w||=1$
  - Maximize margin ρ $\Leftrightarrow$ minimize $||w||$

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 \text{ subject to } r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right) \geq +1, \forall t$$

Margin and support vector

- Support vectors: points lying on the marginal hyperplanes
- NO change of solution does if: remove all other points and retrain
- Margin

$$\min_t \frac{r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right)}{\|\mathbf{w}\|} = \frac{1}{||\mathbf{w}||}$$

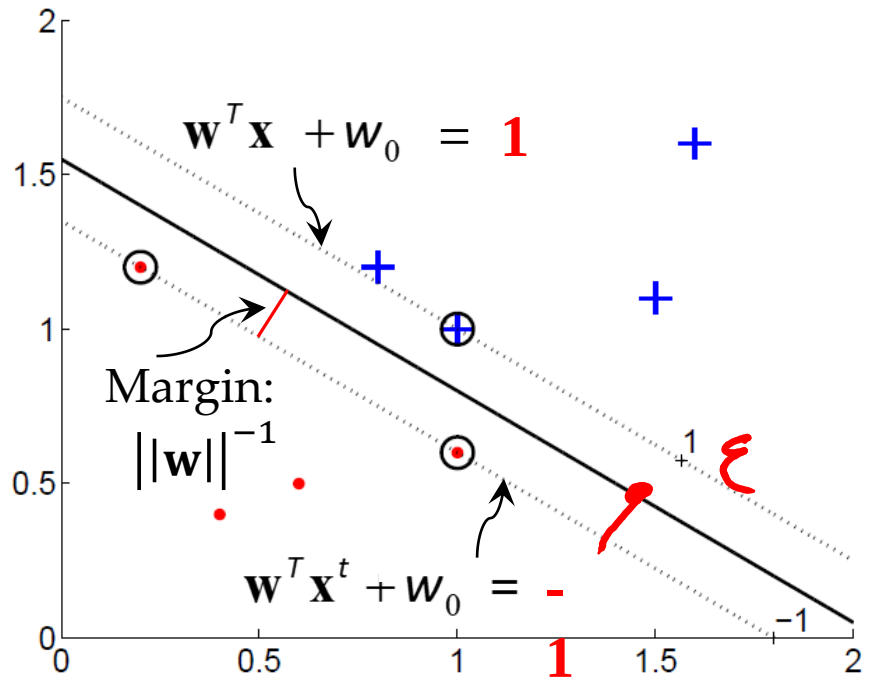
- Marginal hyperplanes

$$\mathbf{w}^T\mathbf{x}^t + w_0 = -1$$

$$\mathbf{w}^T\mathbf{x}^t + w_0 = 1$$

- Separating hyperplane

$$\mathbf{w}^T\mathbf{x}^t + w_0 = 0$$



Soft Margin Hyperplane

- Linear separable:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 \text{ subject to } r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right) \geq +1, \forall t$$

- Not linearly separable

$$r^t\left(\mathbf{w}^T x^t + w_0\right) \geq 1 - \xi^t$$

- Soft error $\sum_t \xi^t$
- New (primal) objective is

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_t \xi^t \quad \text{subject to } r^t\left(\mathbf{w}^T x^t + w_0\right) \geq 1 - \xi^t \qquad \xi^t \geq 0$$

high values of C
lower tolerances for misclassified points

Soft Margin Classification Mathematically

The old formulation:

Find $\mathbf{w}$ such that
$\Phi(\mathbf{w}) = \mathbf{w}^T\mathbf{w}$ is minimized
and for all $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i(\mathbf{w}^T\mathbf{x}_i) \geq 1$

Modified formulation incorporates slack variables:

hinge loss

$\sum w^2$

Find $\mathbf{w}$ such that
$\Phi(\mathbf{w}) = \mathbf{w}^T\mathbf{w} + C\Sigma\xi_i$ is minimized
and for all $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i(\mathbf{w}^T\mathbf{x}_i) \geq 1 - \xi_i$, $\xi_i \geq 0$

Parameter C can be viewed as a way to control overfitting: it "trades off" the relative importance of maximizing the margin and fitting the training data.

- Support vectors: $r^t\left(w^T x^t + w_0\right) \leq 1$
  - Positive points lying on the side of $w^T x^t + w_0 \leq 1$
  - Negative points lying on the side of $w^T x^t + w_0 \geq -1$
  - NO change of solution if:

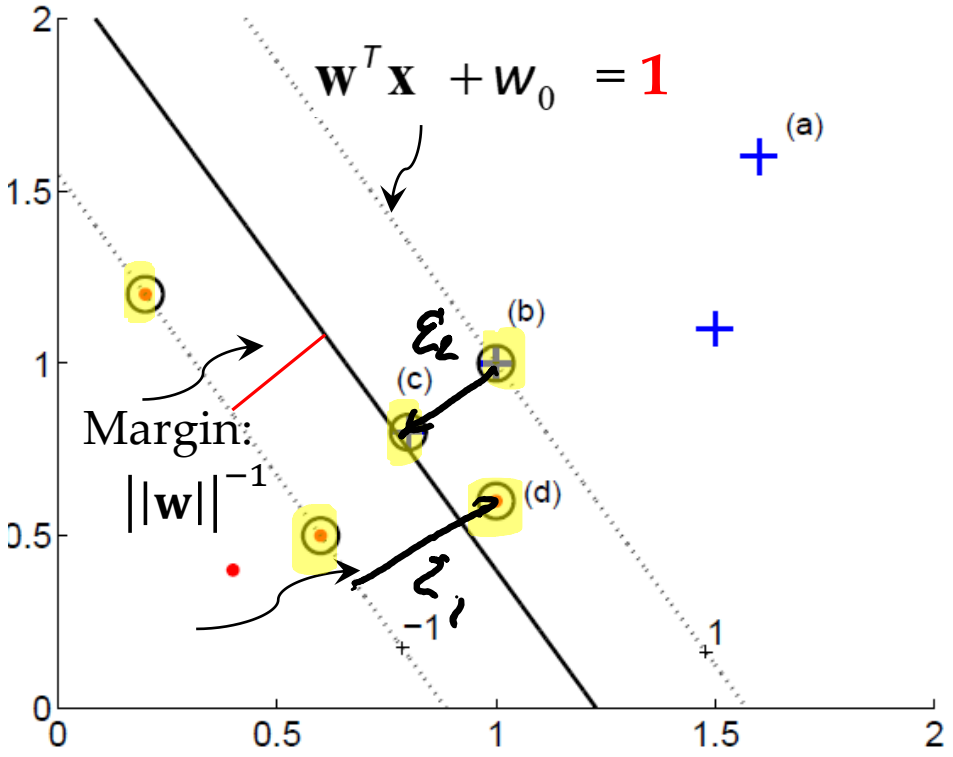
    remove all other points

    and retrain

- Margin? $\dfrac{1}{||\mathbf{w}||} \neq \min_t \dfrac{r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right)}{\|\mathbf{w}\|}$
- Marginal hyperplanes

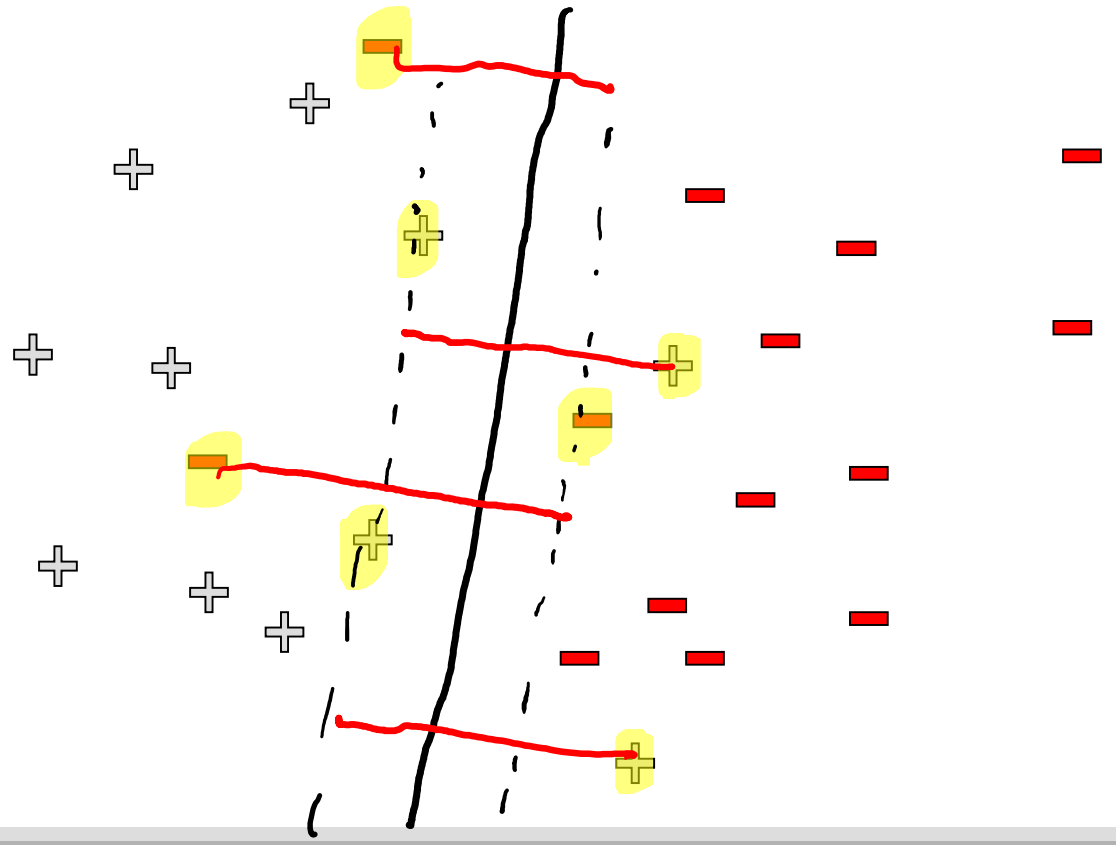
$$\mathbf{w}^T\mathbf{x}^t + w_0 = \mathbf{-1 \text{ or } 1}$$

$$\mathbf{w}^T\mathbf{x}^t + w_0 = \mathbf{-1}$$



Support vectors of SVMs

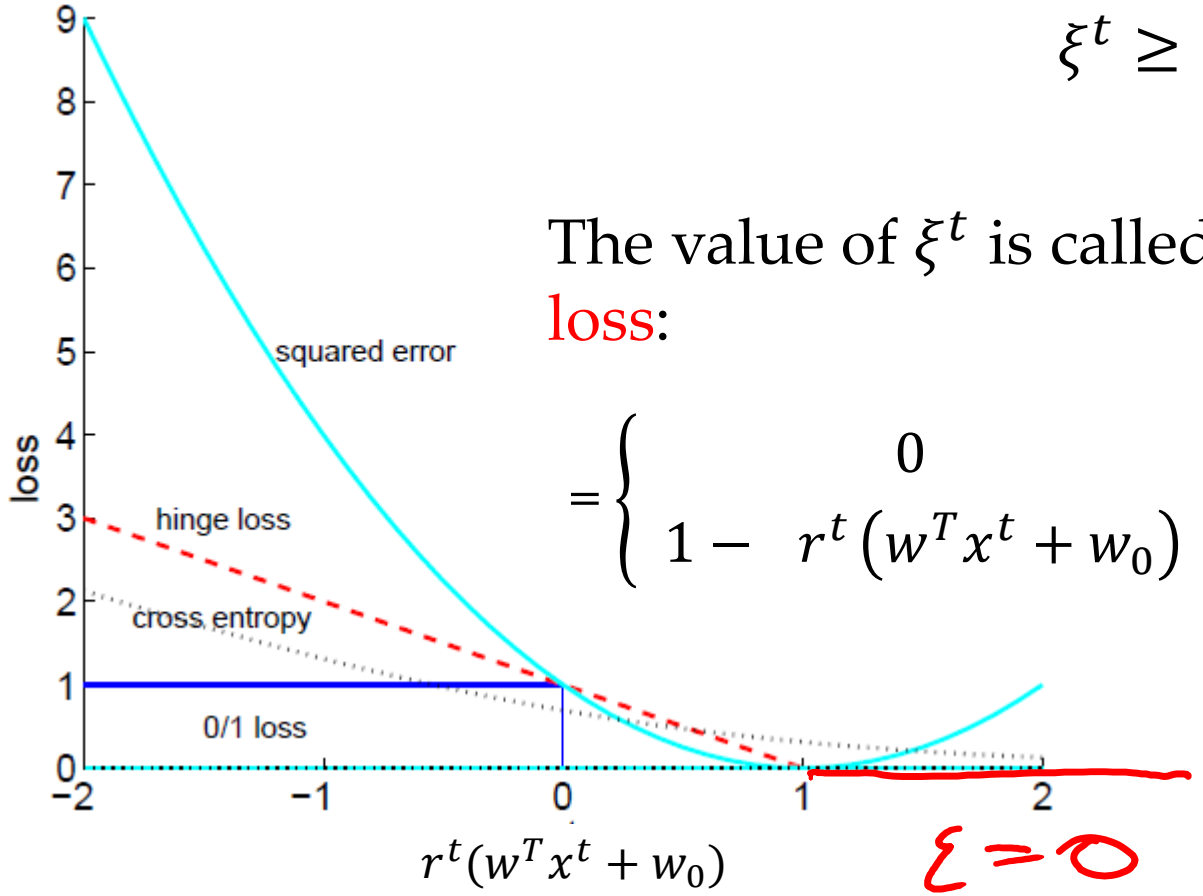
Which examples influence the margin and decision boundaries?



Hinge Loss

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_t \xi^t \quad \text{subject to} \quad r^t\left(\mathbf{w}^T x^t + w_0\right) \geq 1 - \xi^t$$

$$\xi^t \geq 0$$



The value of ξ^t is called hinge loss:

$$= \begin{cases} 0 & \text{if } r^t\left(w^T x^t + w_0\right) \geq 1 \\ 1 - r^t\left(w^T x^t + w_0\right) & \text{otherwise} \end{cases}$$

$\xi = 0$

Linear SVMs: Overview

The classifier is a *separating hyperplane.*

Most "important" training points are support vectors; they define the hyperplane.

Quadratic optimization algorithms can identify which training points $\mathbf{x}_i$ are support vectors with non-zero Lagrangian multipliers α_i.

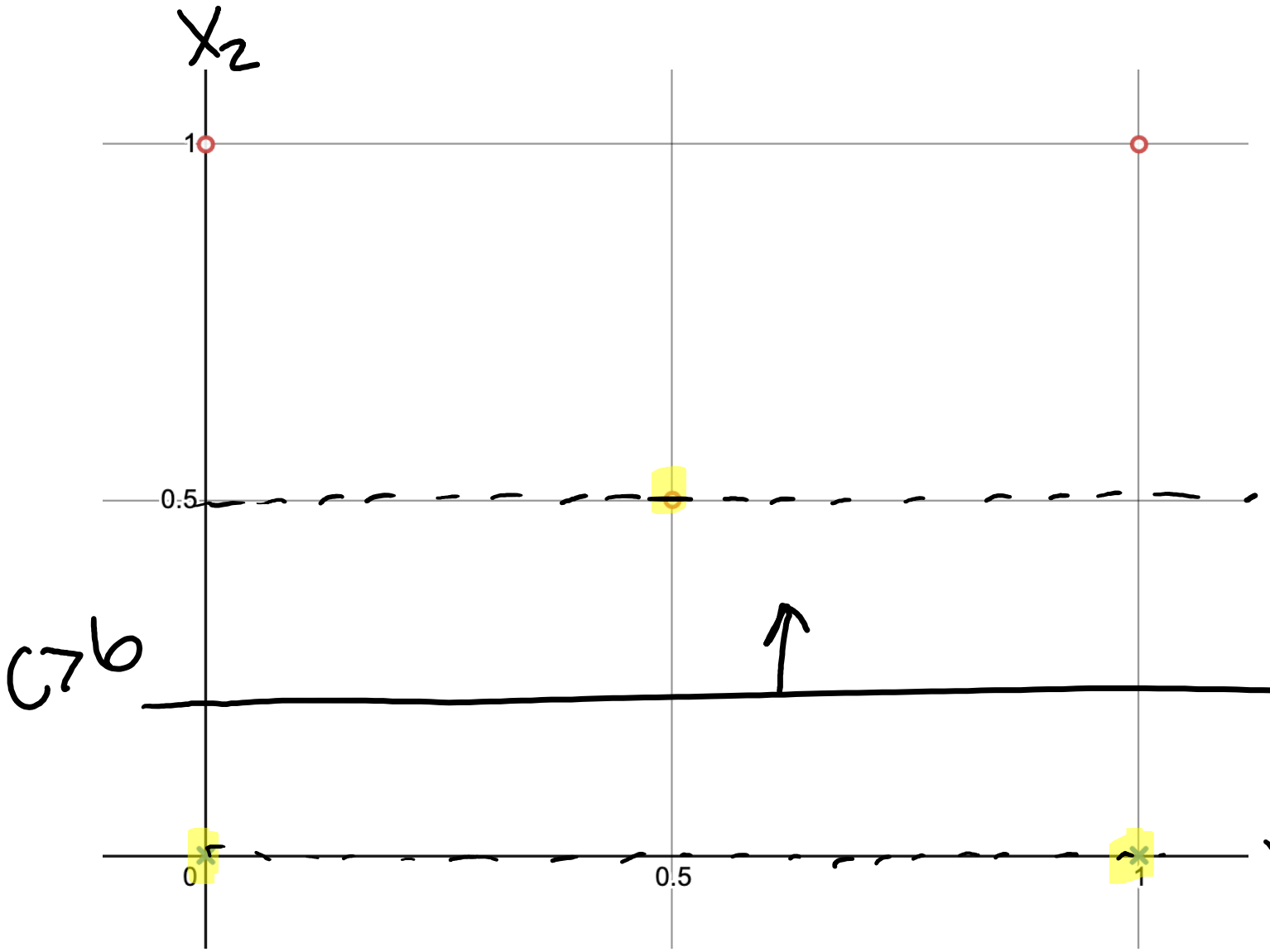
Both in the dual formulation of the problem and in the solution training points appear only inside inner products:

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$ is maximized
and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \le \alpha_i \le C$ for all α_i

$$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^T\mathbf{x} + b$$

α is non zero for support vectors

C from our hyper parameters



$w_0 = -1$

$w_1 = 0$

$w_2 = 4$

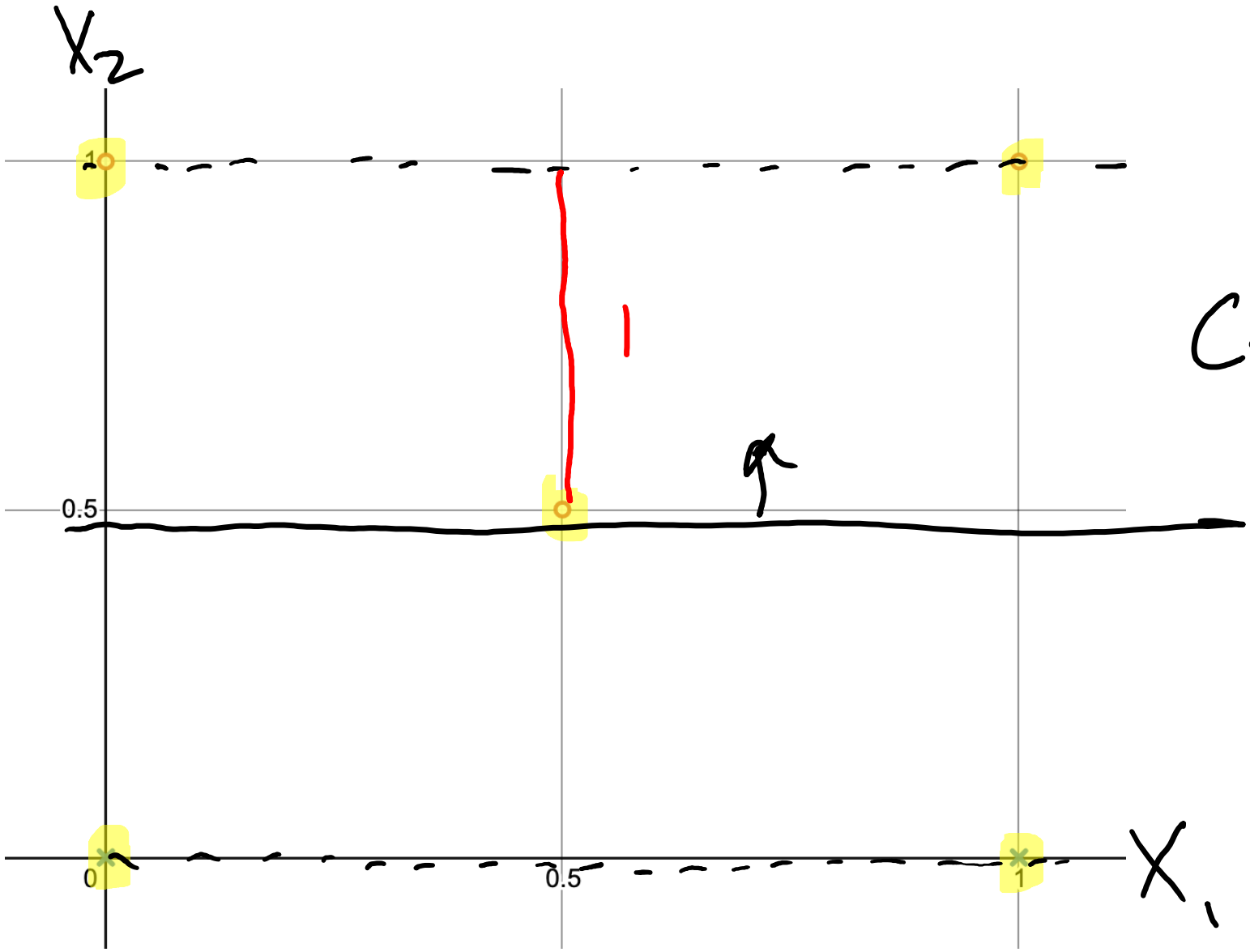
What is w for the hard margin separator?

$w_0 + w_2 0 = -1$

$w_0 + w_2 0.5 = 1$

$w_2 = 2.2 =$

$-1 + \frac{w_2}{2} = 1$



$w_0 = -1$

$w_1 = 6$

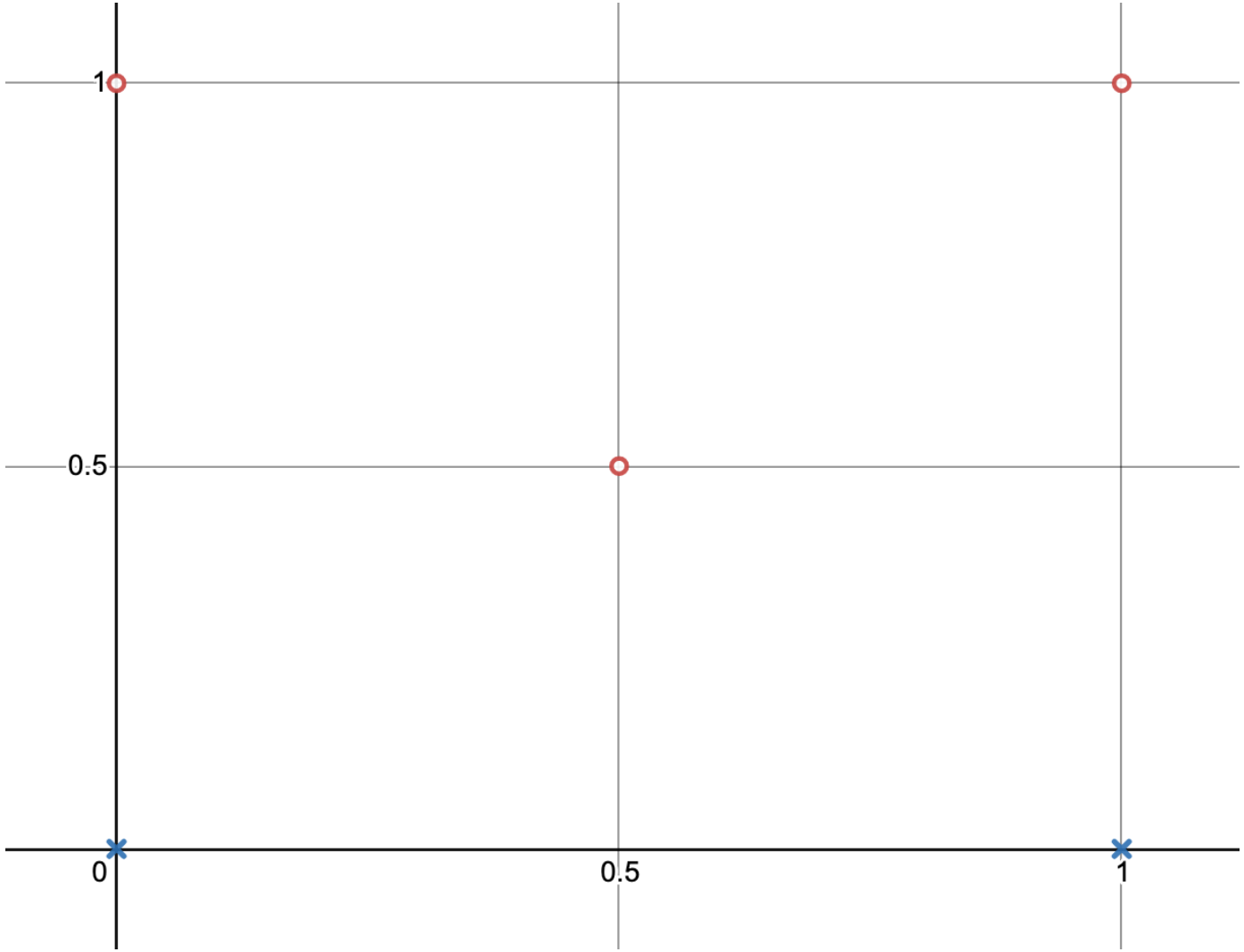
$w_2 = 2$

$c < 6$

What is w for the soft margin separator?

$w_2 \cdot 1 + w_0 = 1$

$w_2 \cdot 0 + w_0 = -1$



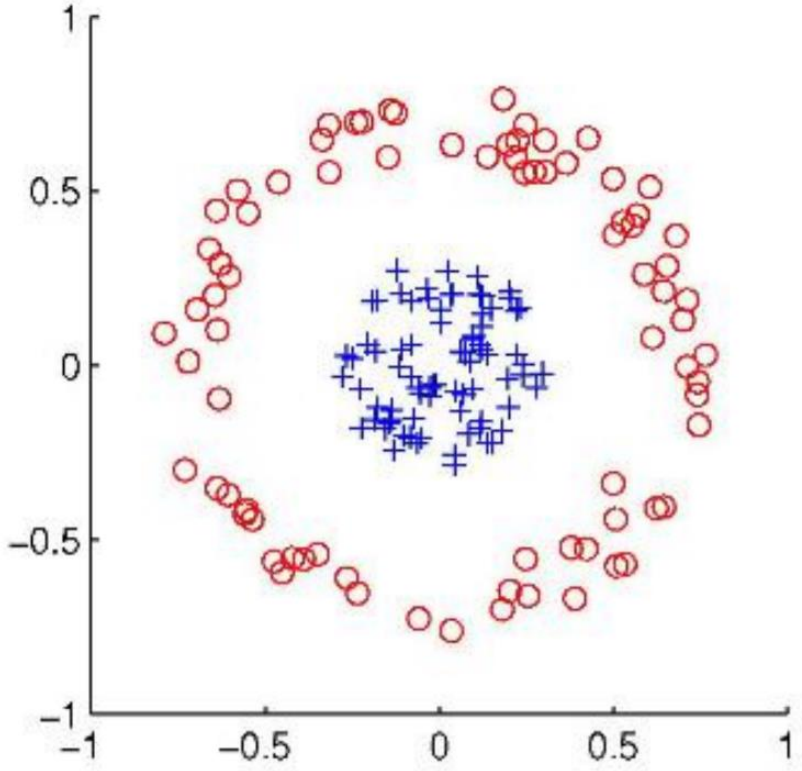
What is the value of C at which the model transitions from one to the other?

$\frac{\|w\|_2}{2} + C \cdot \varepsilon$

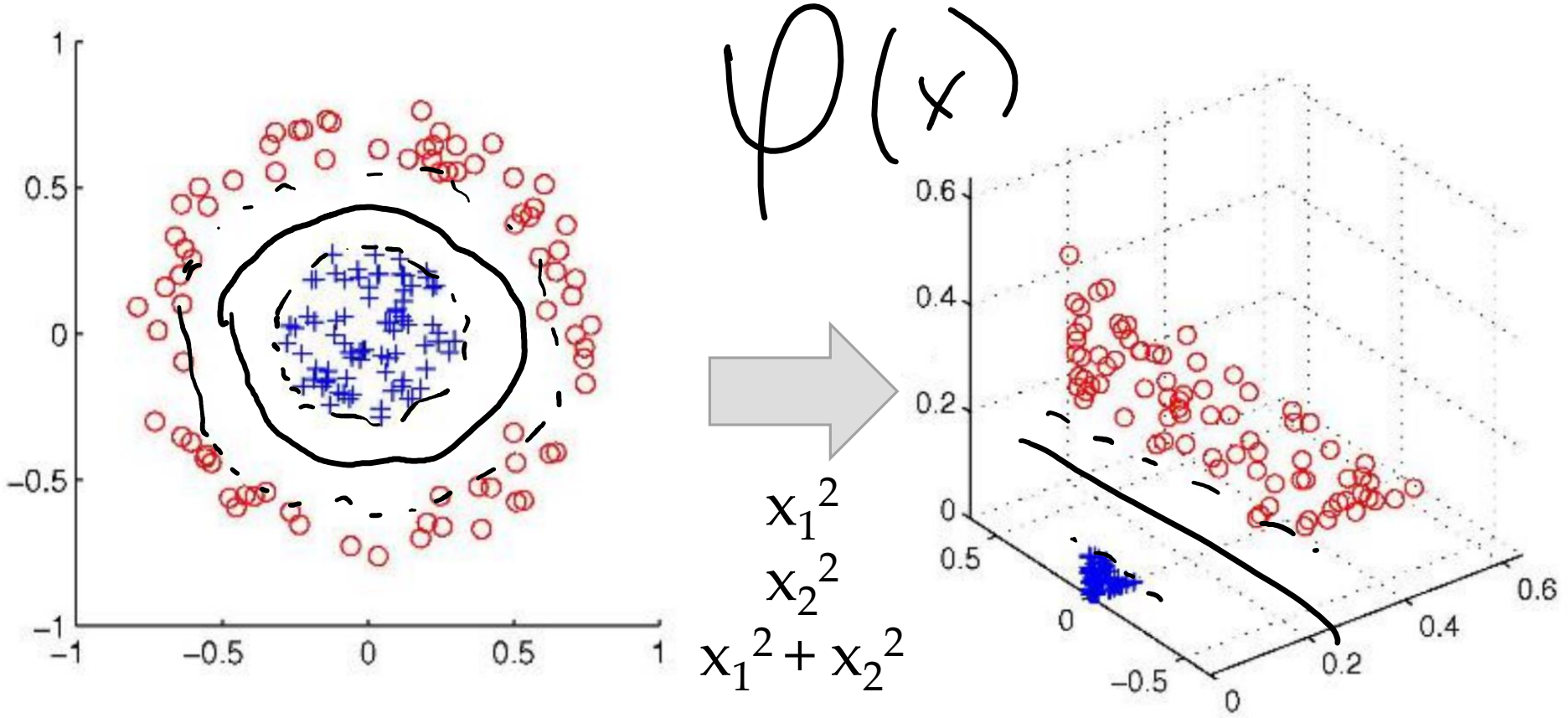
$17 = 5 + 2C \qquad C = 6$

$\frac{17}{2} + C \cdot 0 = \frac{5}{2} + C \cdot 1$

What if the data is not linearly separable?

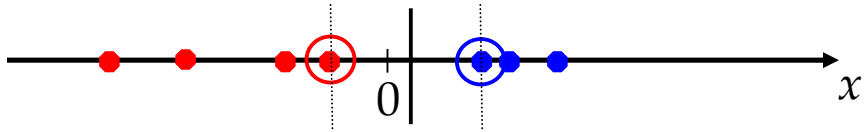


What if the data is not linearly separable?

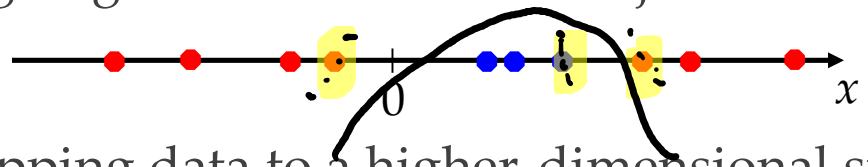


Non-linear SVMs

Datasets that are linearly separable with some noise work out great:

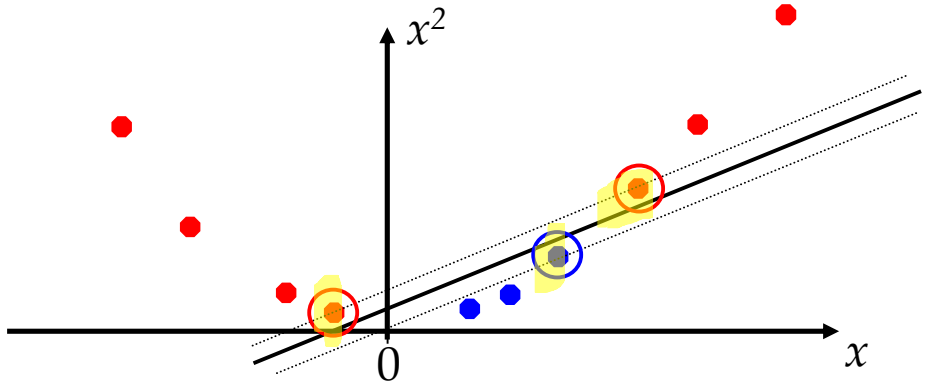


But what are we going to do if the dataset is just too hard?



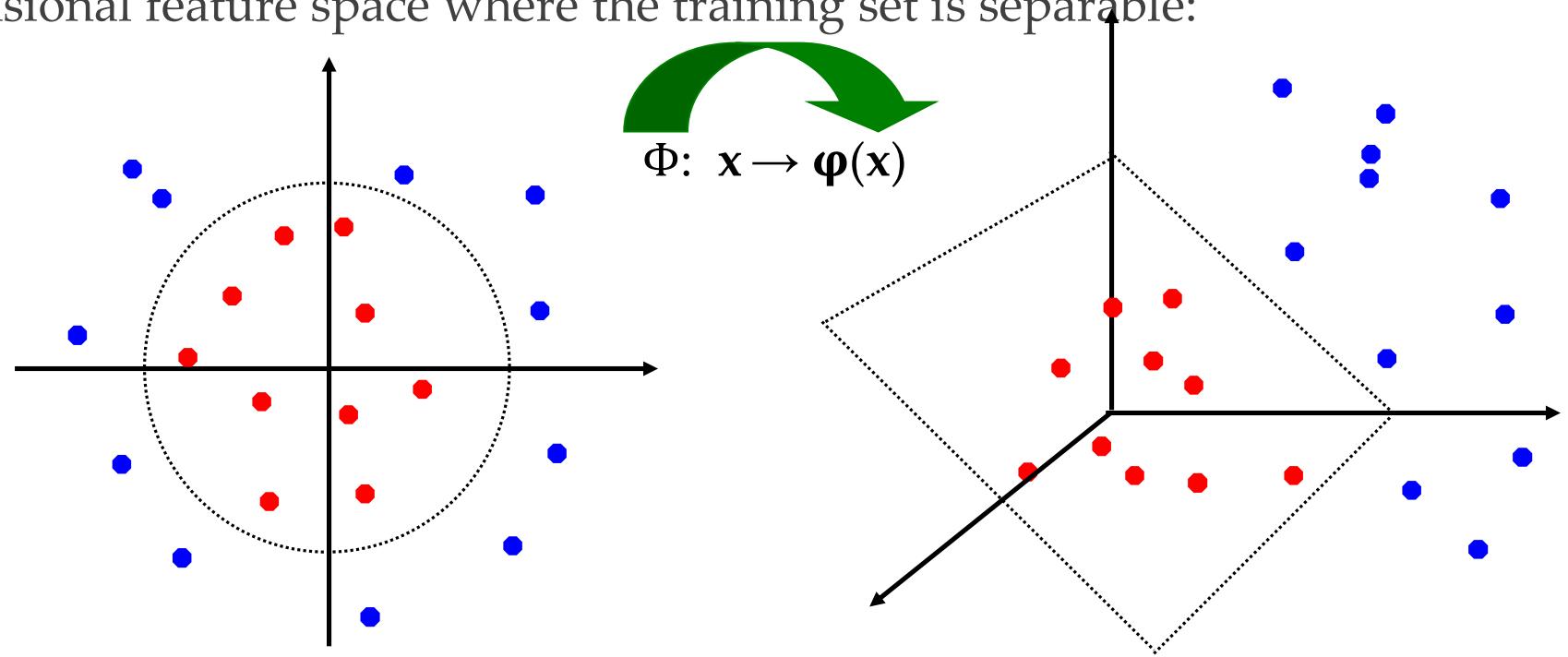
How about… mapping data to a higher-dimensional space:

degree-2 polynomial kernel



Non-linear SVMs: Feature spaces

General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:



Kernel Trick

- Preprocess input x by basis functions

$$z = \varphi(x) \qquad g(z)=w^T z$$

$$g(x)=w^T \varphi(x)$$

- The SVM solution

$$\mathbf{w} = \sum_t \alpha^t r^t \mathbf{z}^t = \sum_t \alpha^t r^t \boldsymbol{\varphi}\left(\mathbf{x}^t\right)$$

$$g(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) = \sum_t \alpha^t r^t \boldsymbol{\varphi}\left(\mathbf{x}^t\right)^T \boldsymbol{\varphi}(\mathbf{x})$$

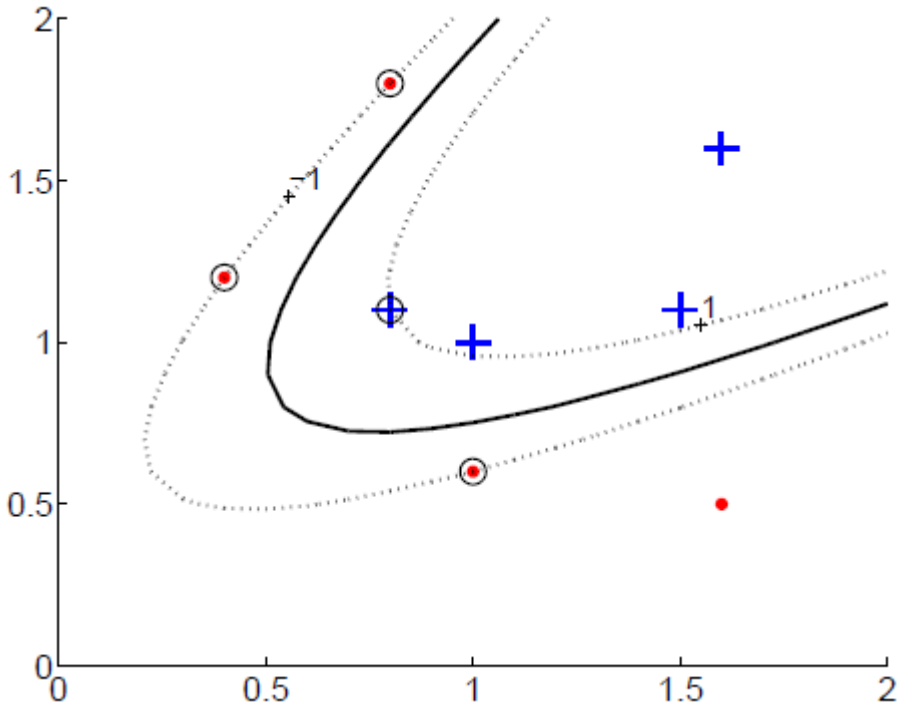
$$g(\mathbf{x}) = \sum_t \alpha^t r^t K\left(\mathbf{x}^t, \mathbf{x}\right)$$

Vectorial Kernels

- Polynomials of degree q:

$$K(\mathbf{x}^t, \mathbf{x}) = (\mathbf{x}^T \mathbf{x}^t + 1)^q$$

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + 1)^2$$
$$= (x_1 y_1 + x_2 y_2 + 1)^2$$
$$= 1 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 x_2 y_1 y_2 + x_1^2 y_1^2 +$$
$$\phi(\mathbf{x}) = \left[1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1^2, x_2^2\right]^T$$



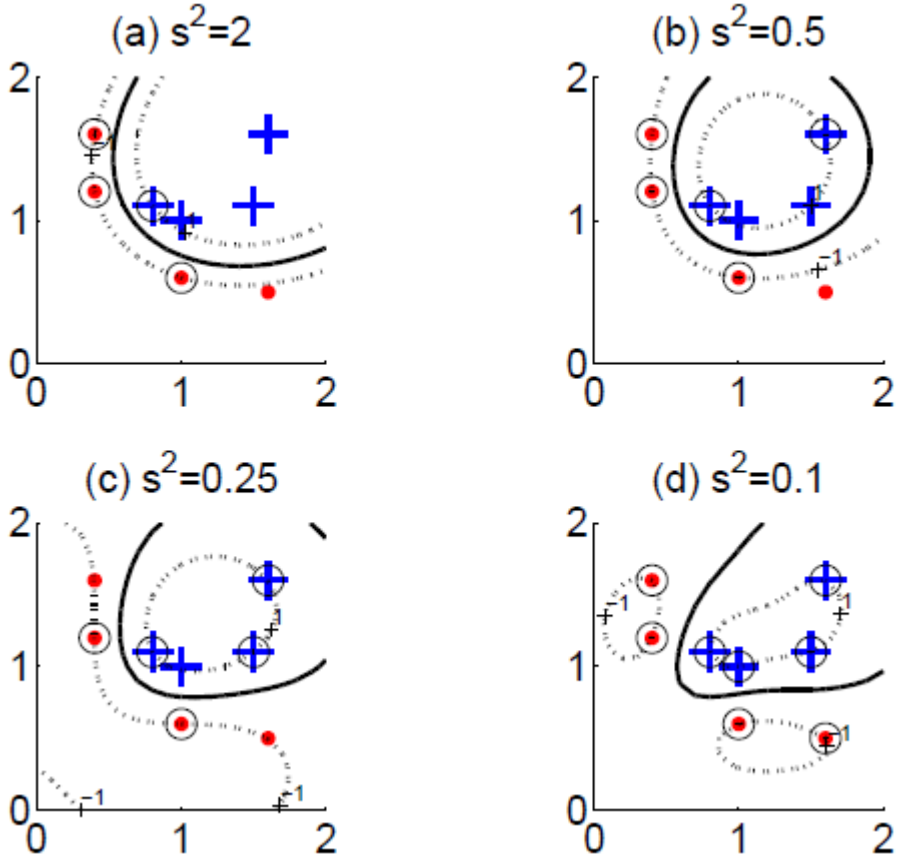
$x_1, x_2 \Rightarrow \{$ $\}$

Vectorial Kernels

- Radial-basis functions:

$$K(\mathbf{x}^t, \mathbf{x}) = \exp\left[-\frac{\|\mathbf{x}^t - \mathbf{x}\|^2}{2s^2}\right]$$

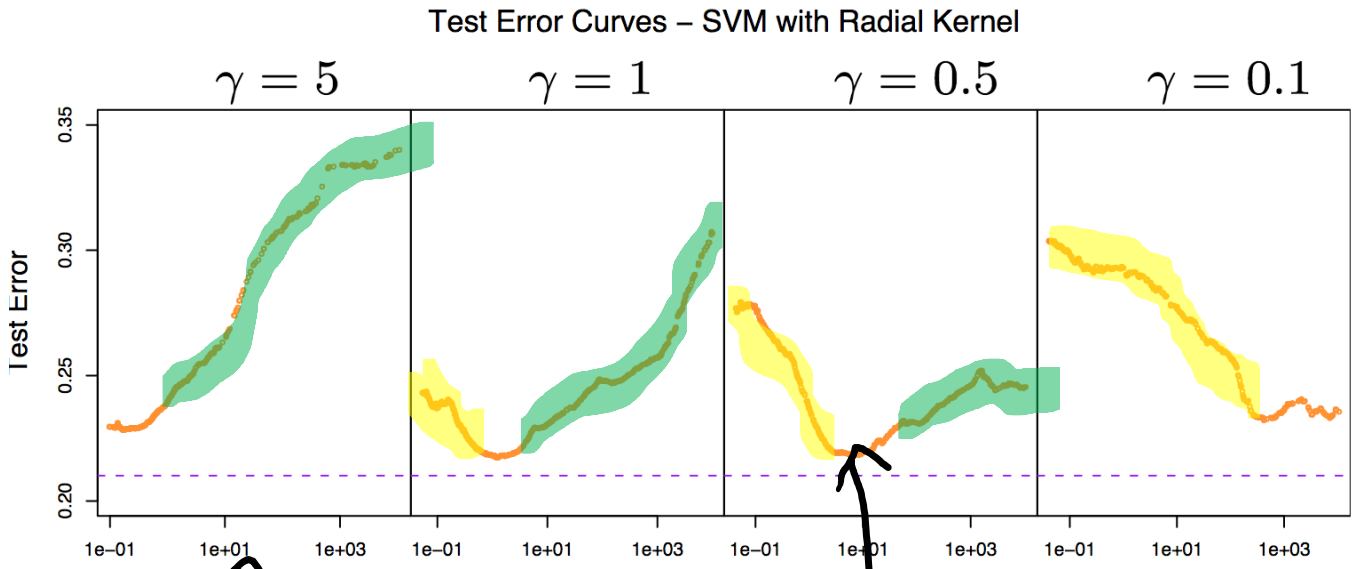
γ

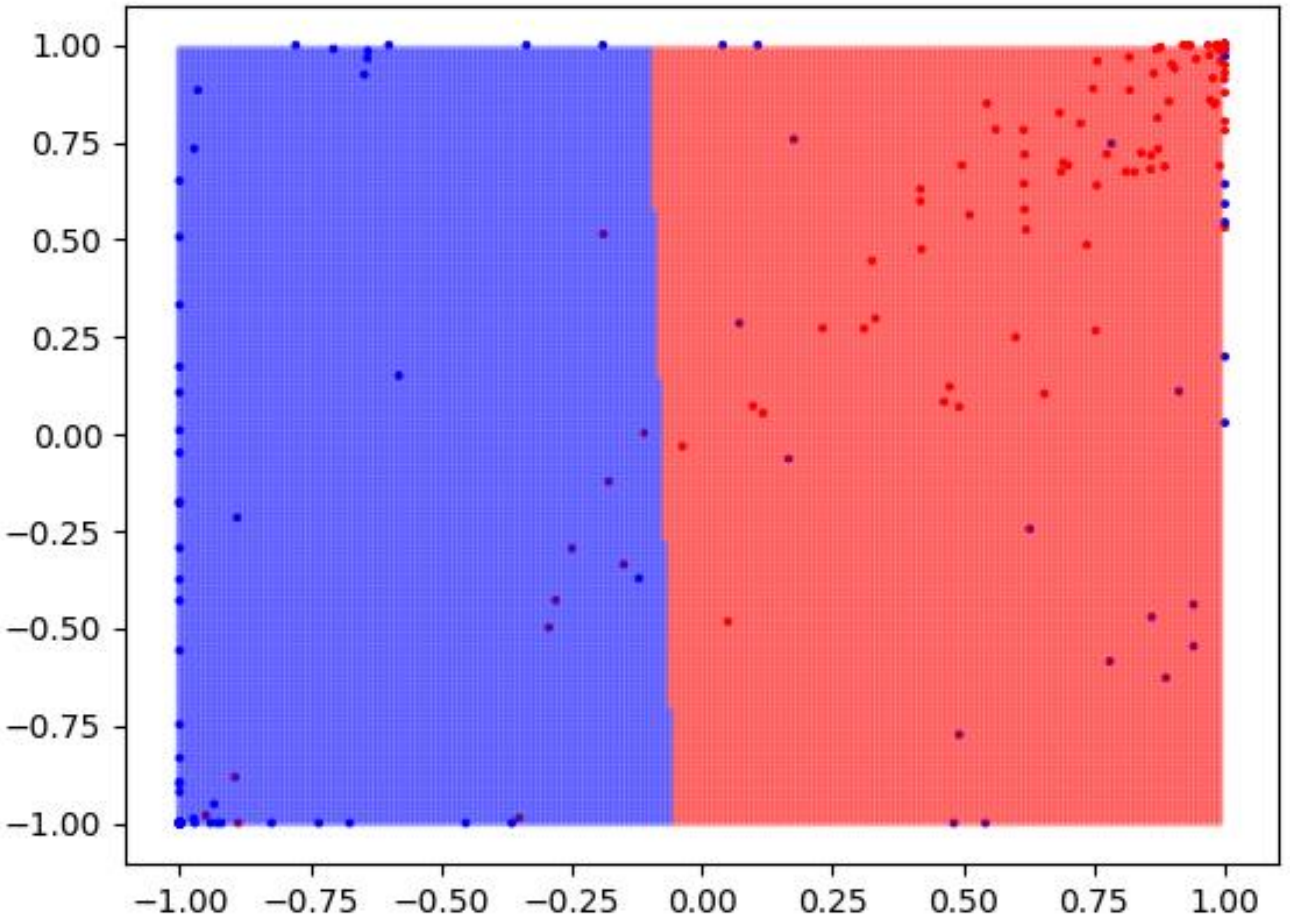


Overfitting

Because of the high dimensionality of the kernel model, there is a stronger need to adjust for overfitting

- Here *c (x-axis)* is the regularization paramater and
- λ is the "scale parameter" for the model which indicates its allowed complexity.

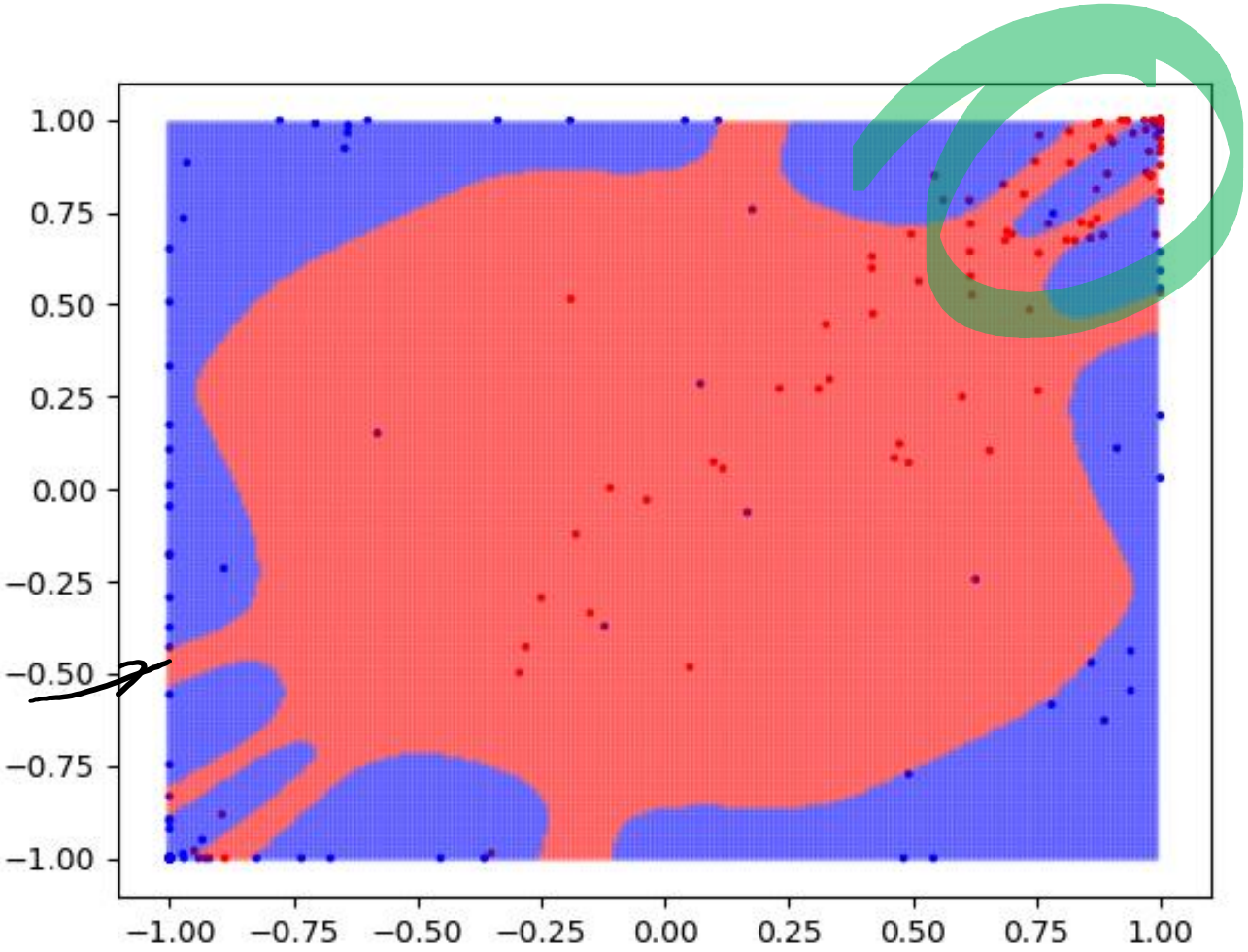




Over/underfitting

Here is the 72/88 pixel data with the soft margin linear model over the top of it

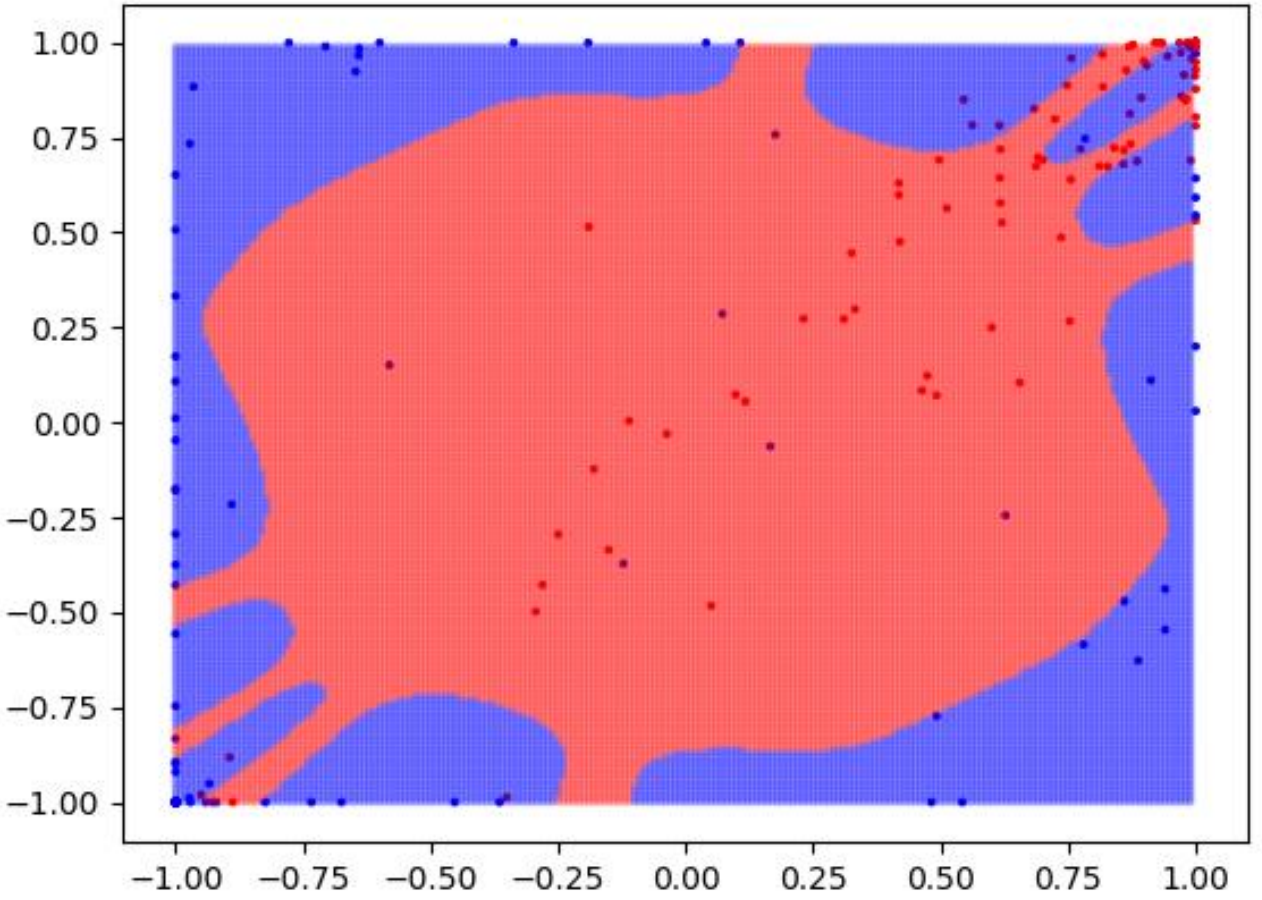
Is this over/underfit?



Over/underfitting

Here is a polynomial kernel, with degree 20

- Does this seem over/underfit?

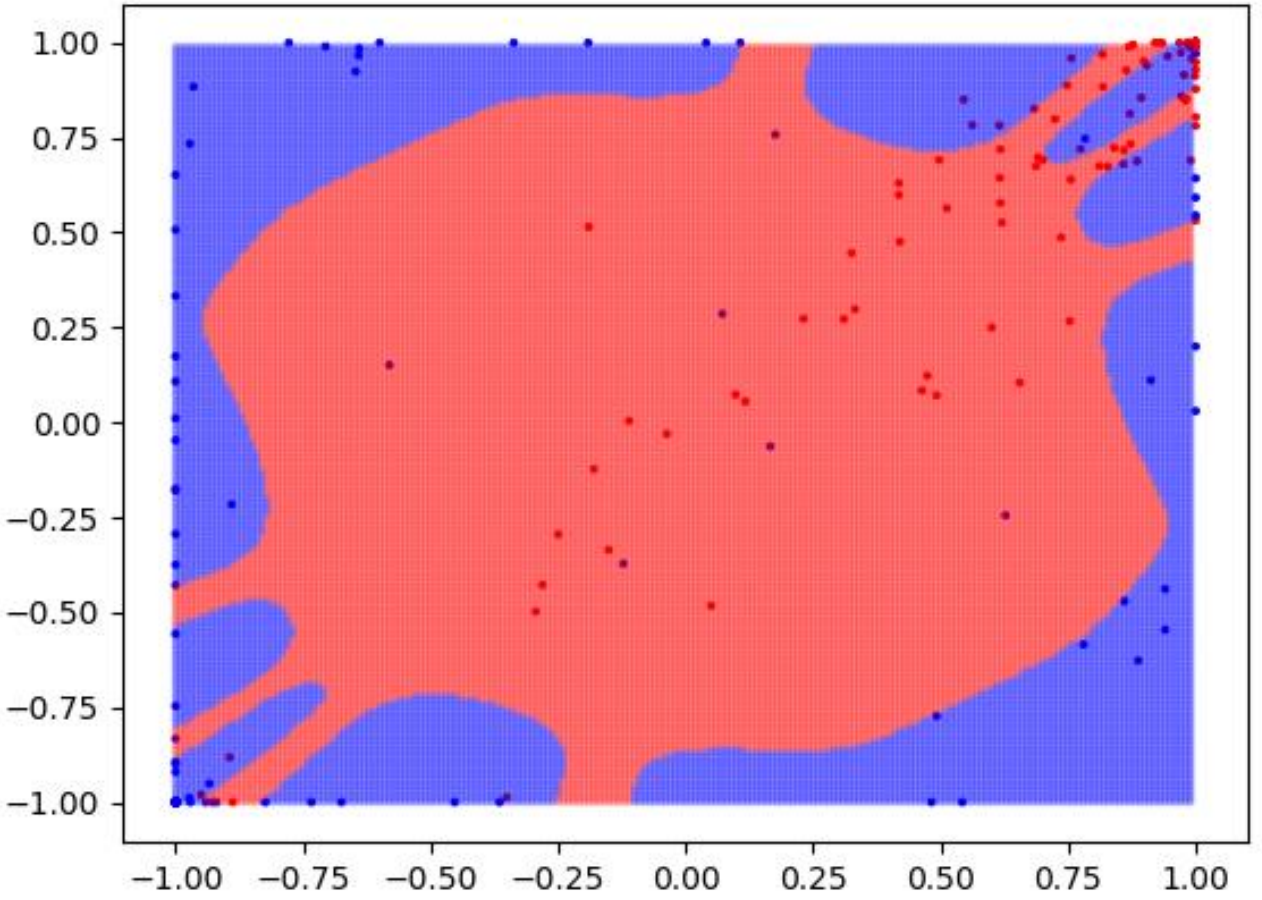


Over/underfitting

Here is a polynomial kernel.

Does this seem over/underfit?

- What is the degree of this kernel? (High/low)?

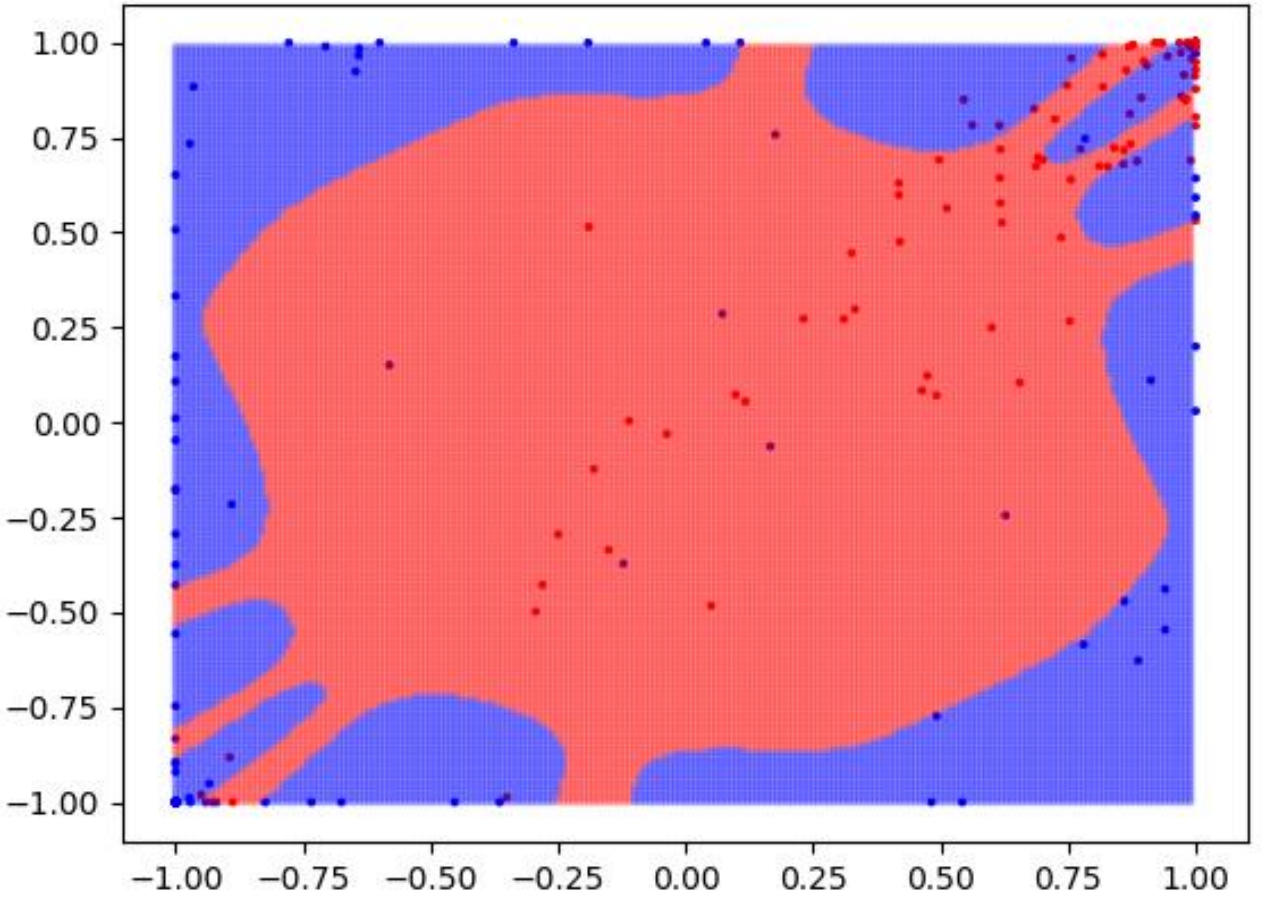


Over/underfitting

Here is a polynomial kernel.

Does this seem over/underfit?

- What is the degree of this kernel? (High/low)?
- What do you think is the value for c?



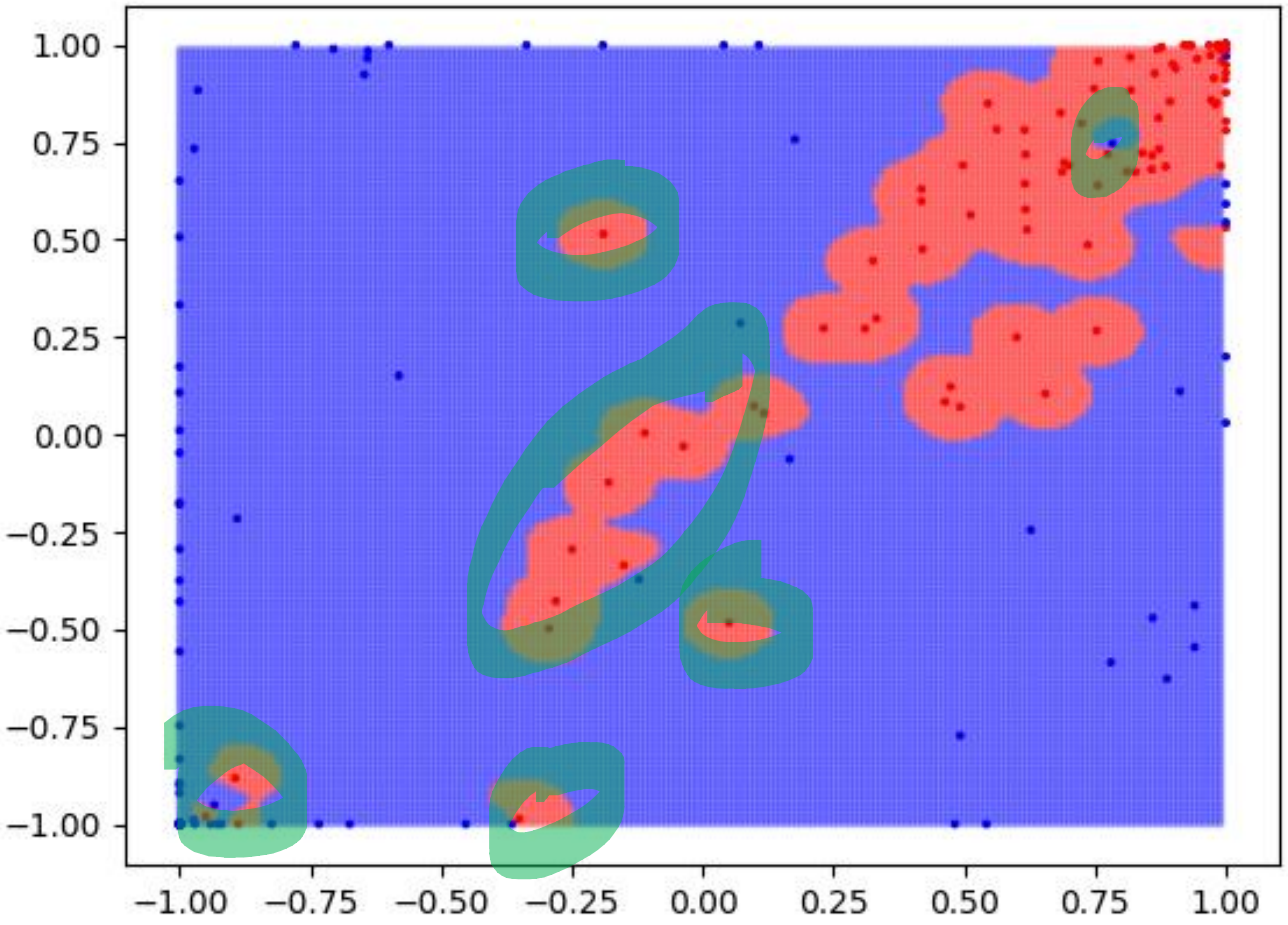
Over/underfitting

Here is a polynomial kernel.

Does this seem over/underfit?

- What is the degree of this kernel? (High/low)?
- What do you think is the value for c?

What about this graph might indicate overfitting visually?

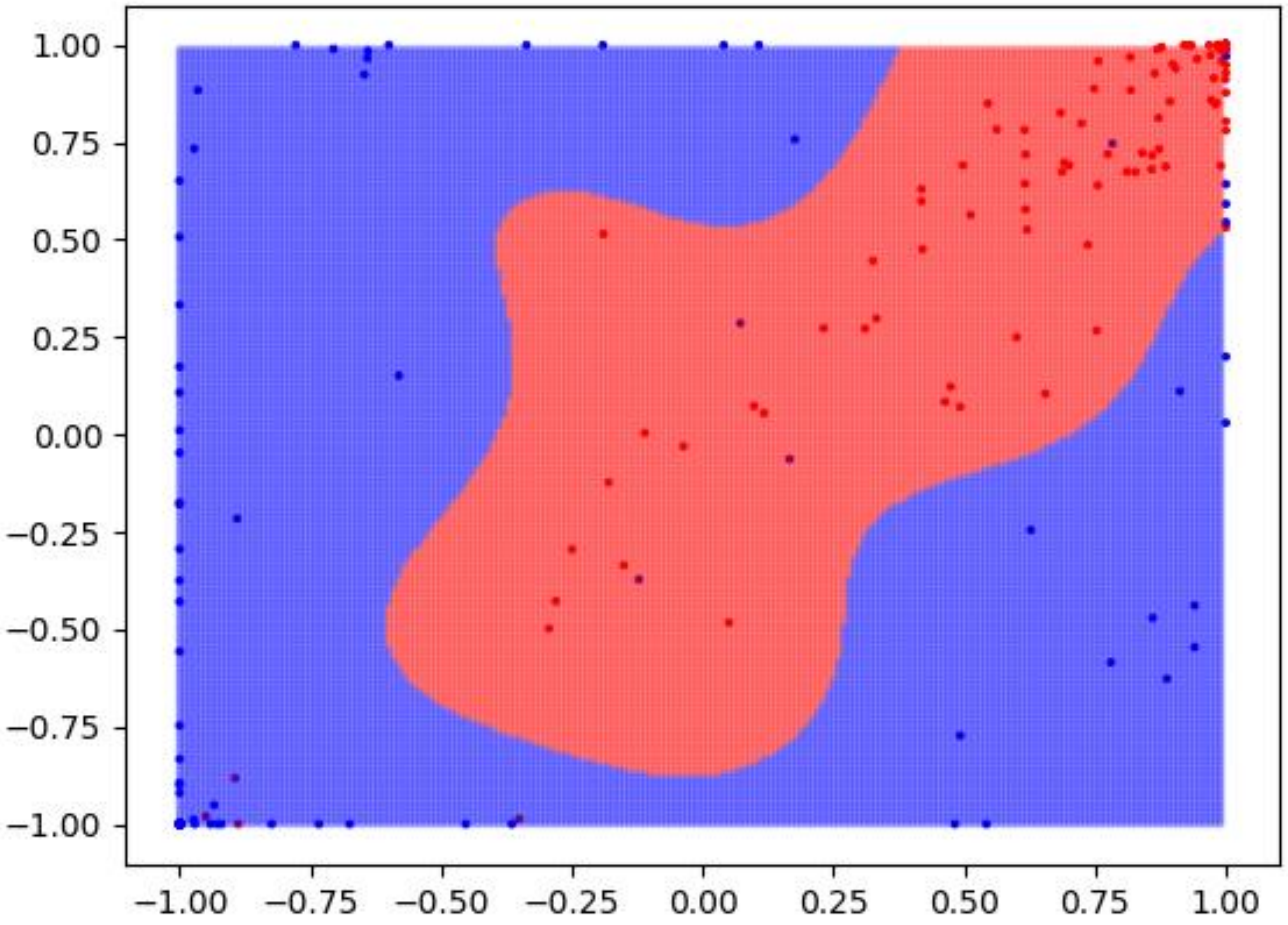


Over/underfitting

Here is a radial kernel

- How complex is the kernel relative to the feature space?

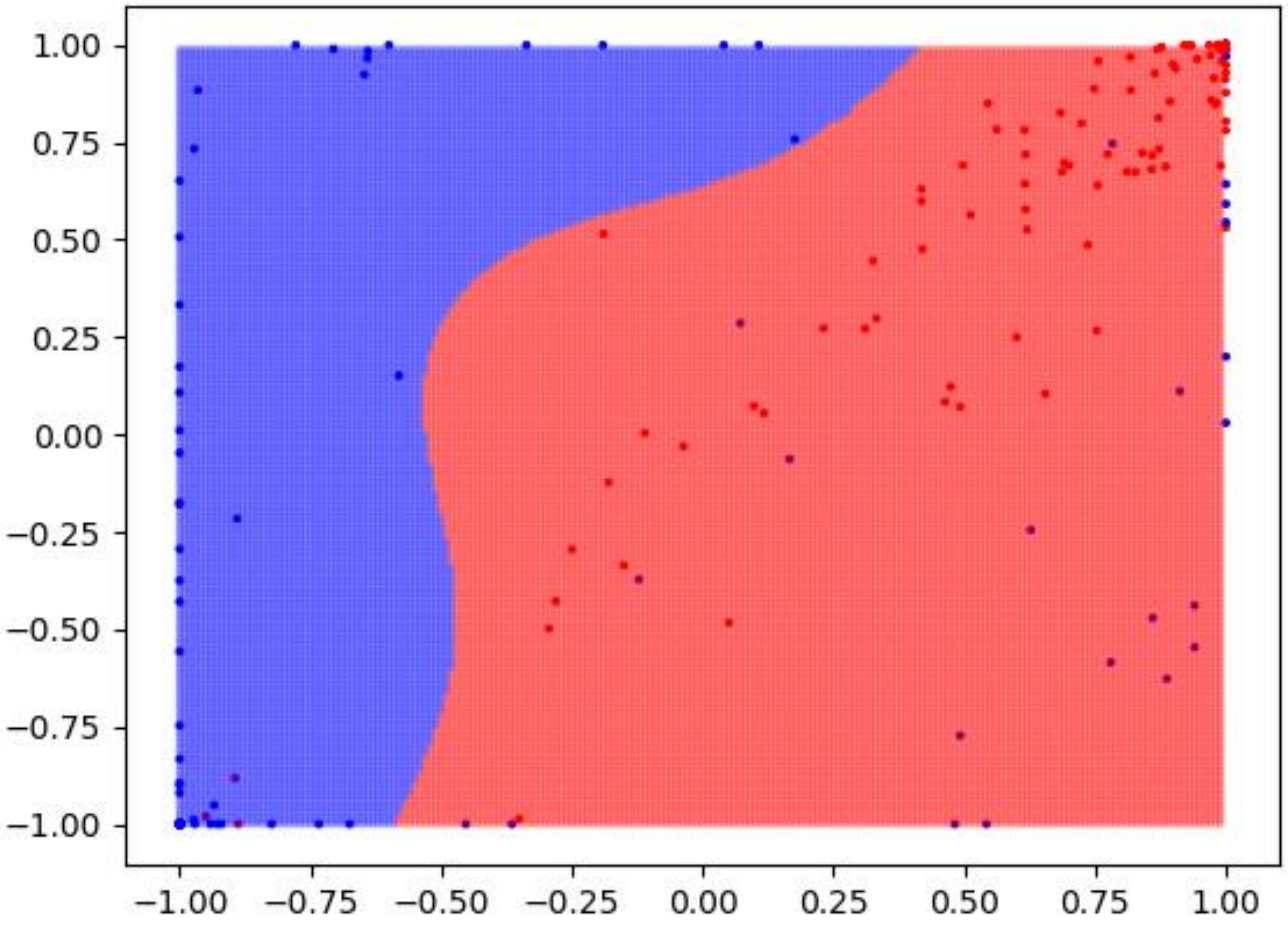
What can you say about the likely cross-validation error for this model?



Over/underfitting

How about this radial model?

- Complexity?
- C?



Over/underfitting

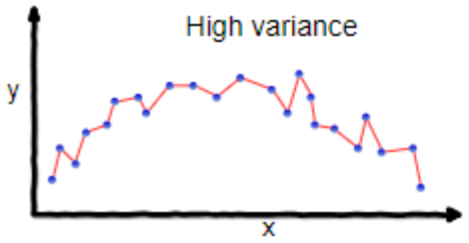
How about this polynomial model?

- Complexity?
- C?

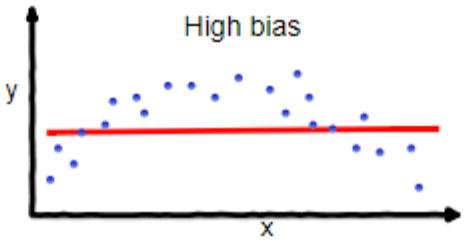
Over/underfitting

Recall the bias vs. variance tradeoff that we we've been discussing so far in the course

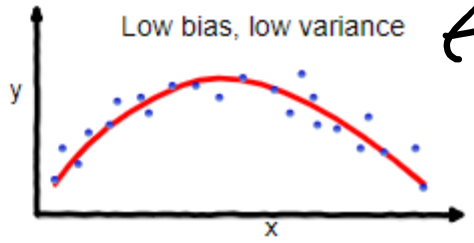
Unlike in kNN, where the model complexity peaked when k = 1,
SVMs have a lot more potential complexity
(and therefore a lot more capacity for overfitting)



overfitting



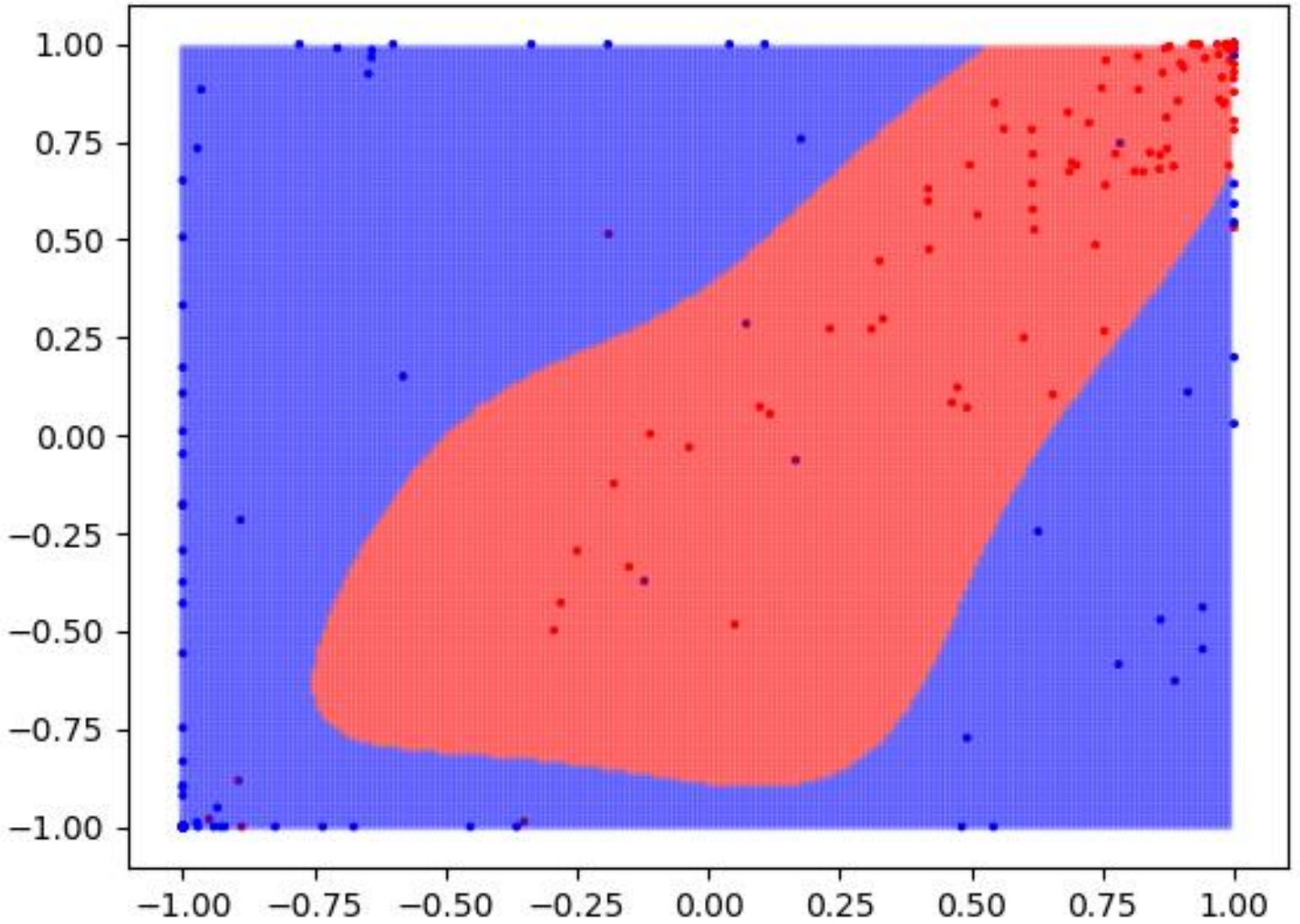
underfitting

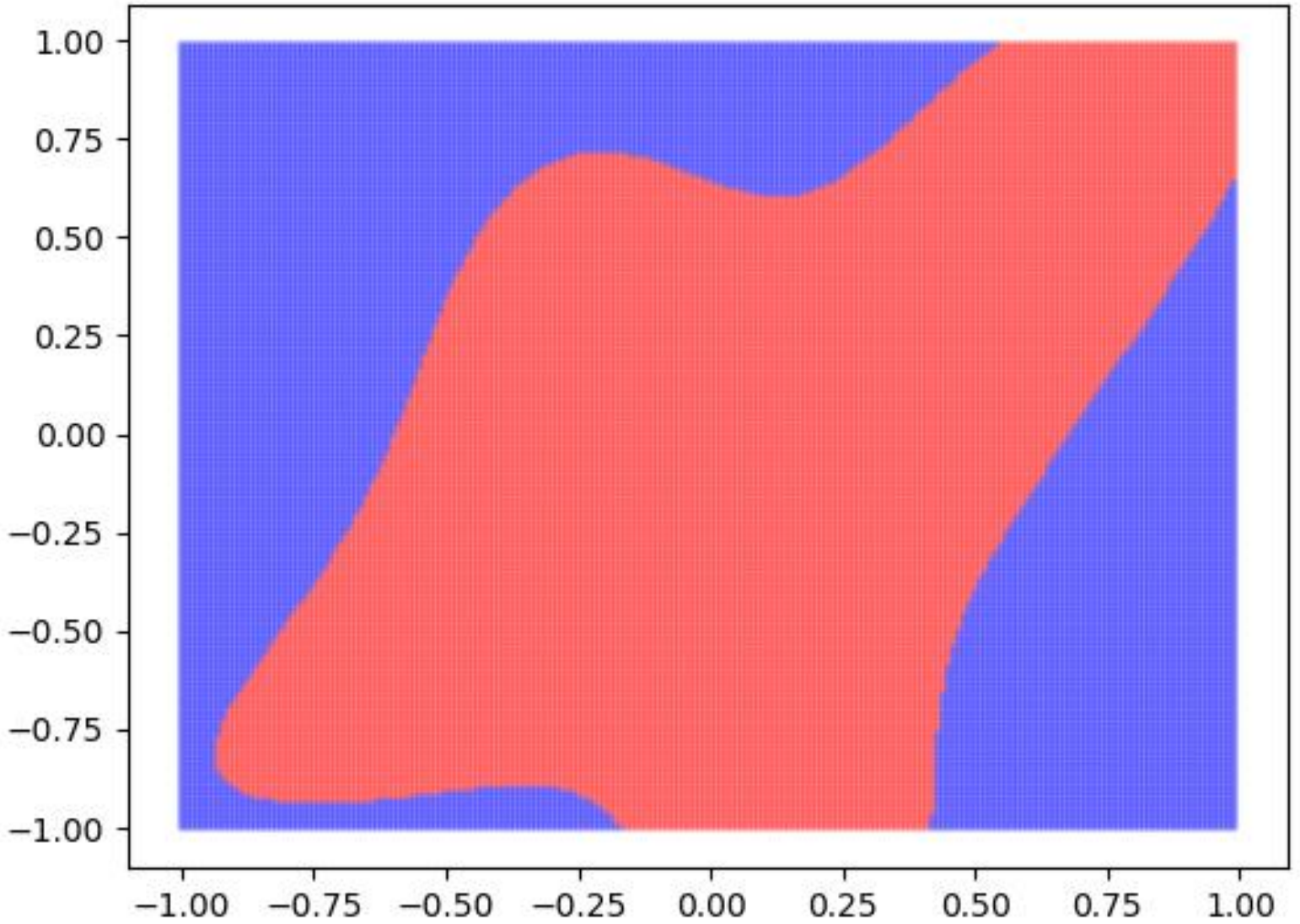


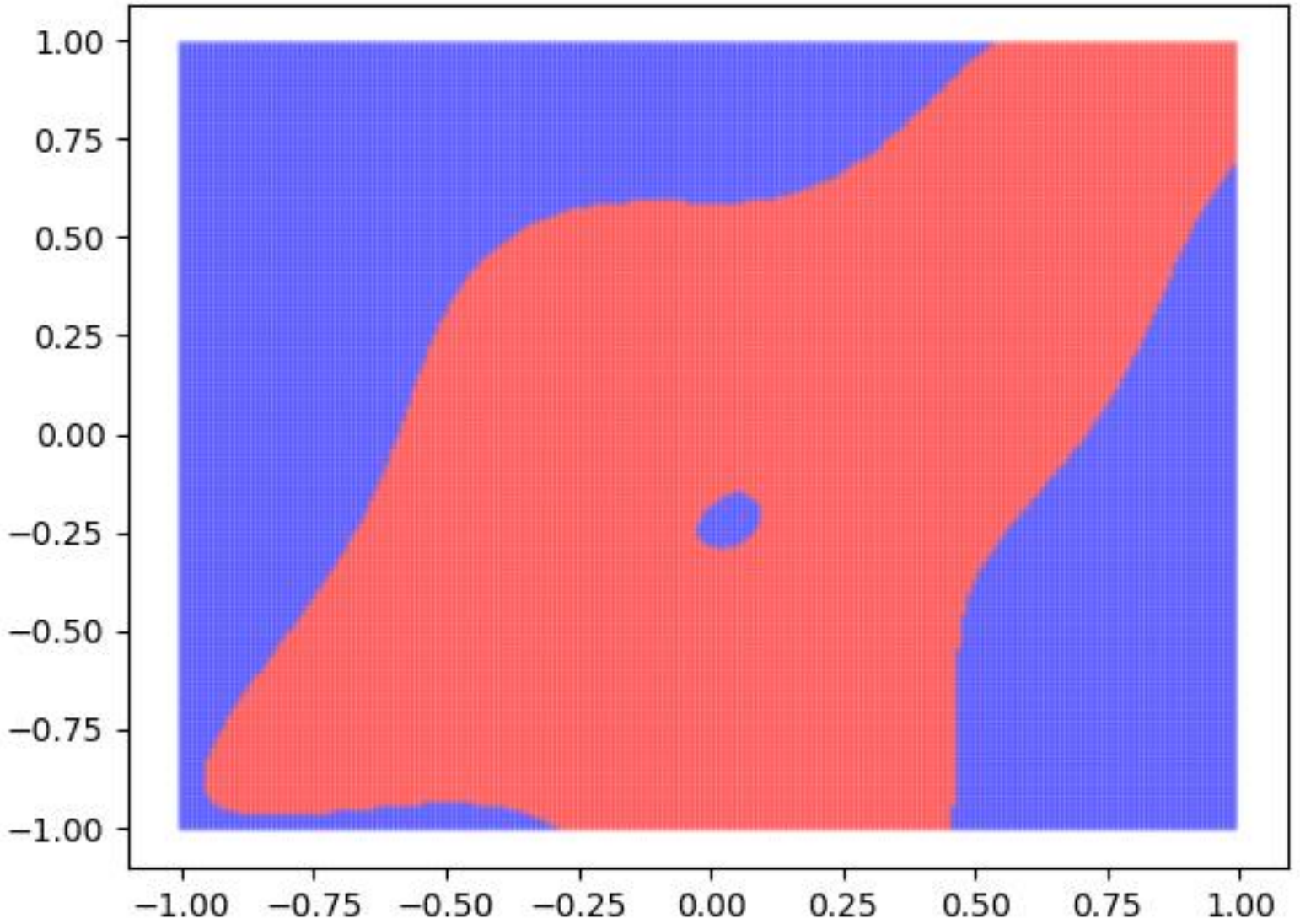
Good balance

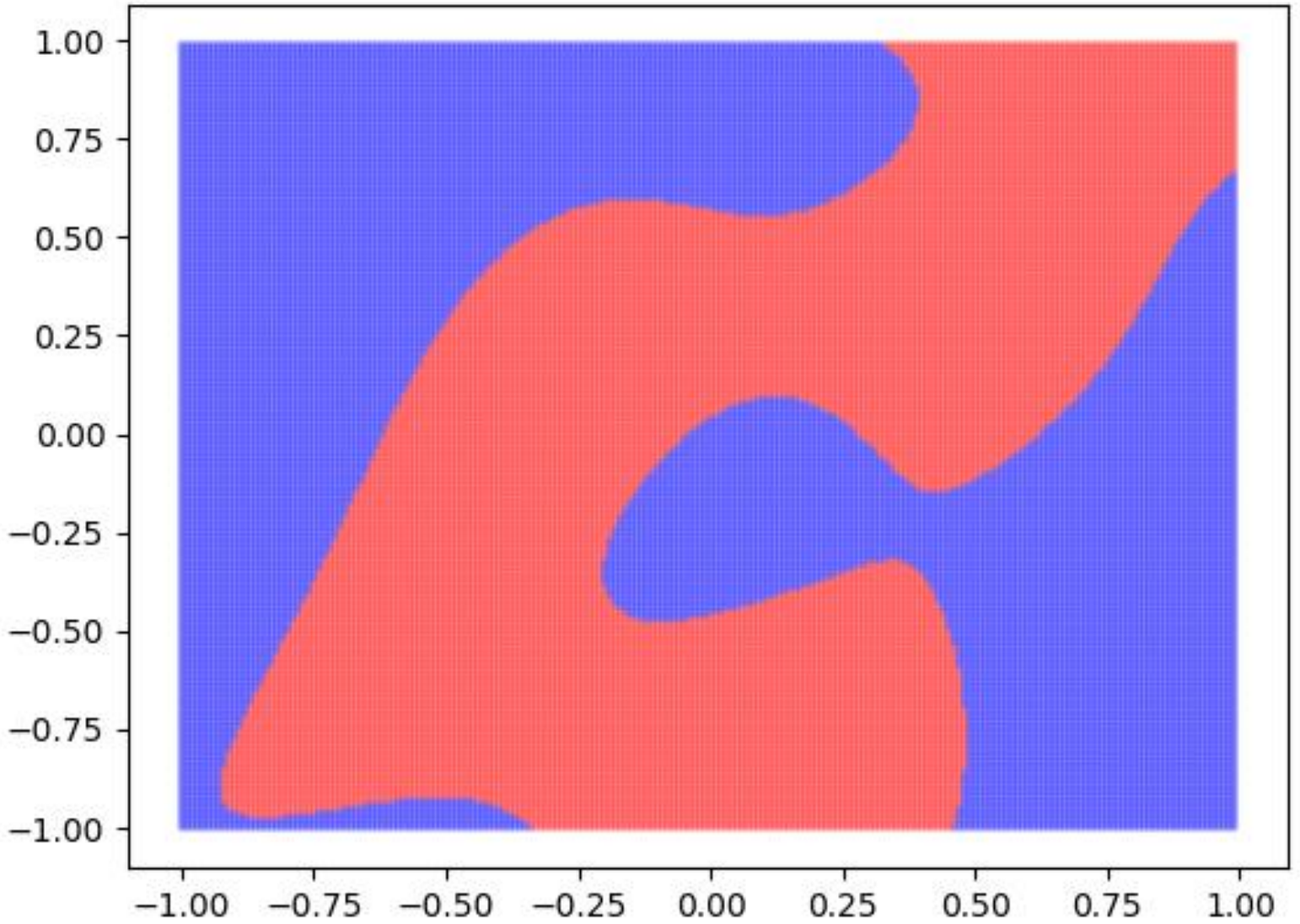
always a tradeoff

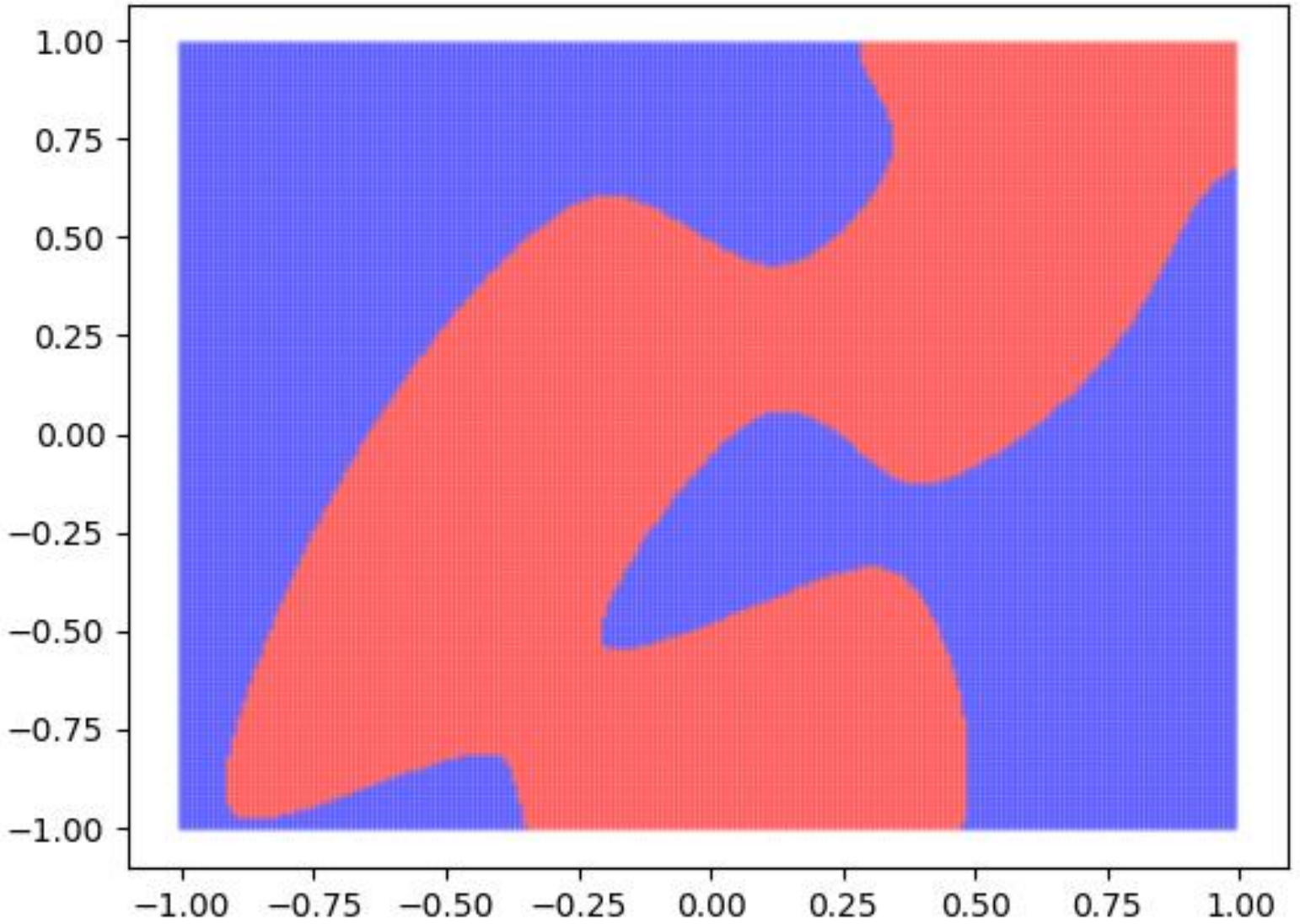
E_{CV}

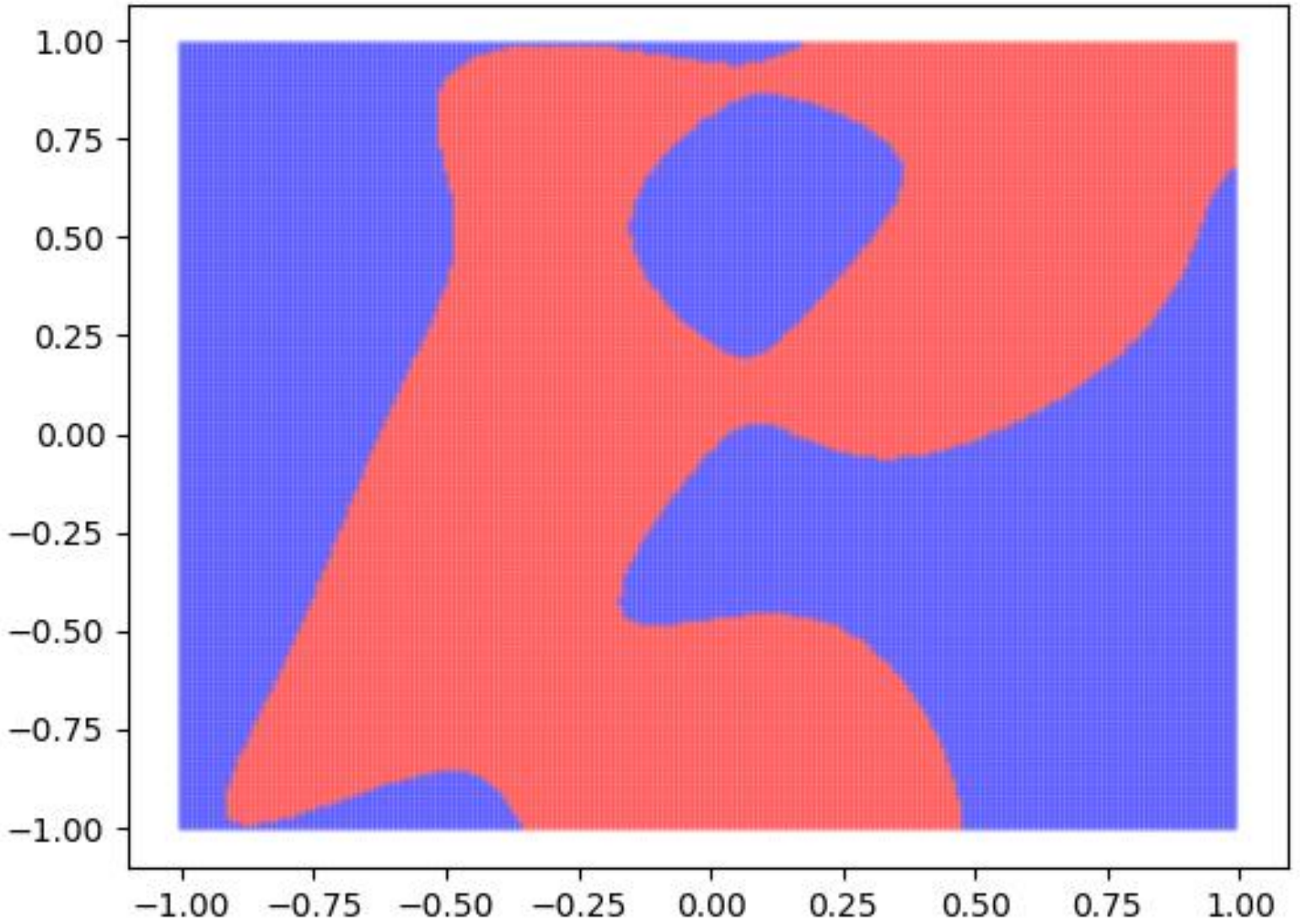

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

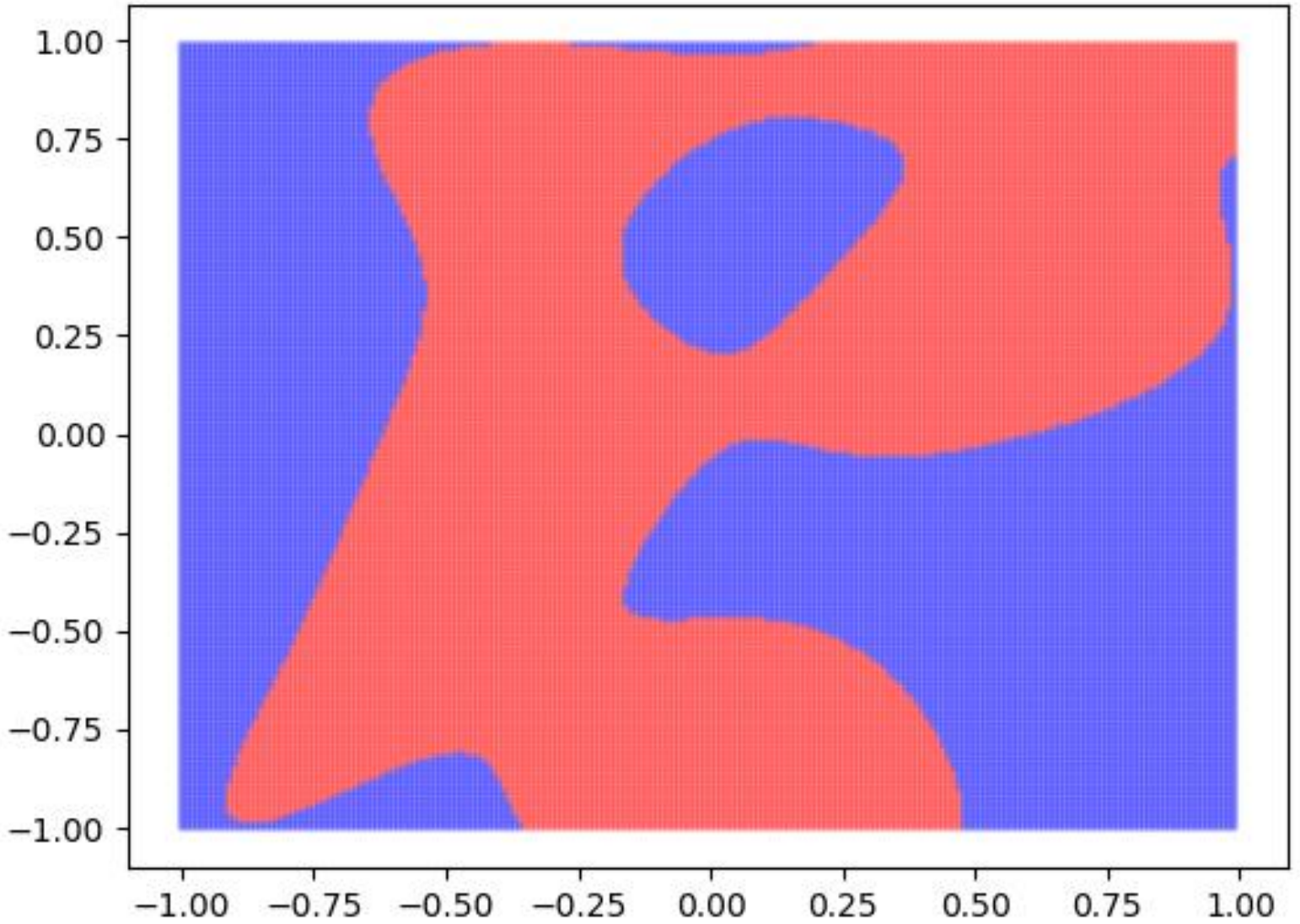

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

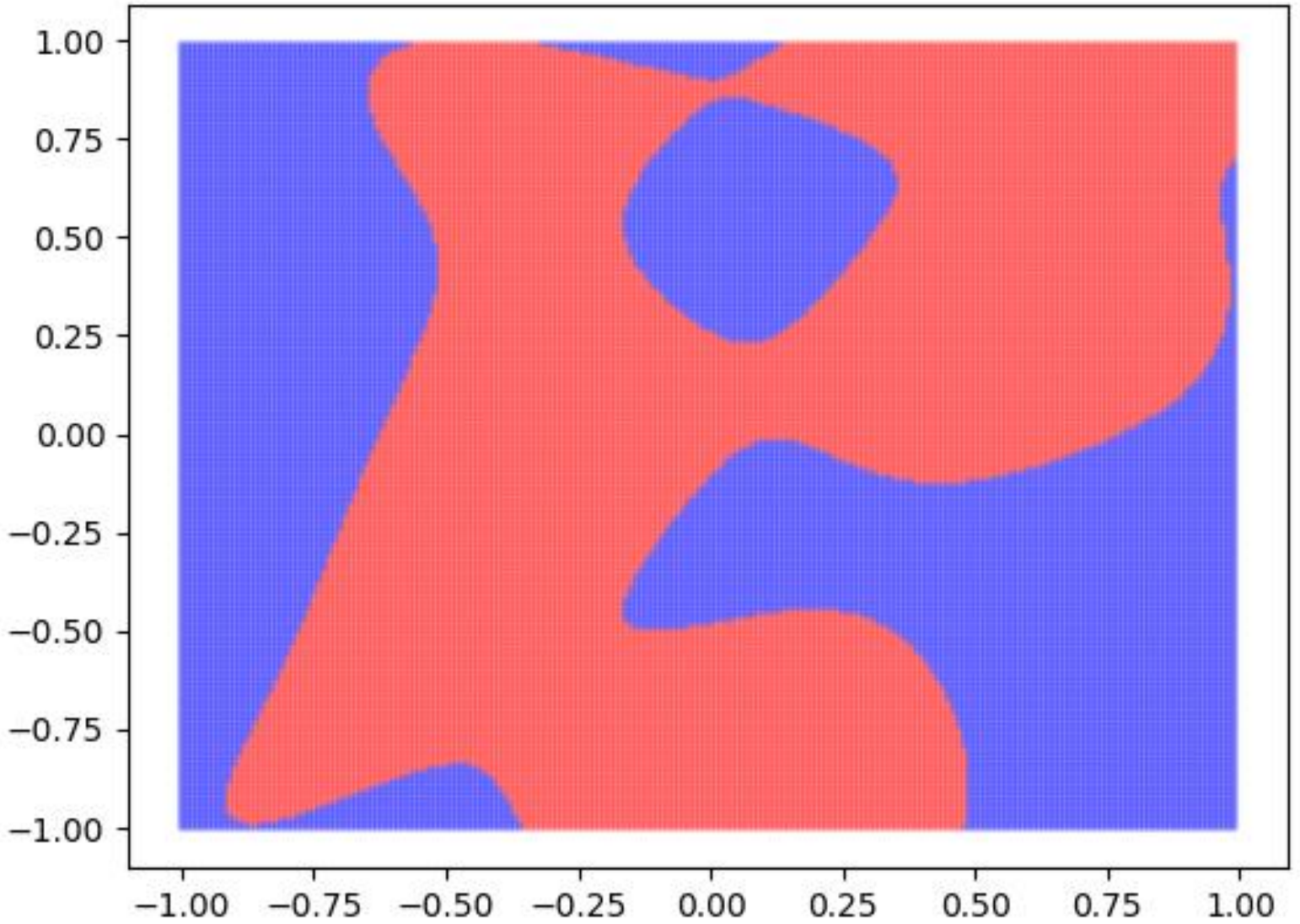

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

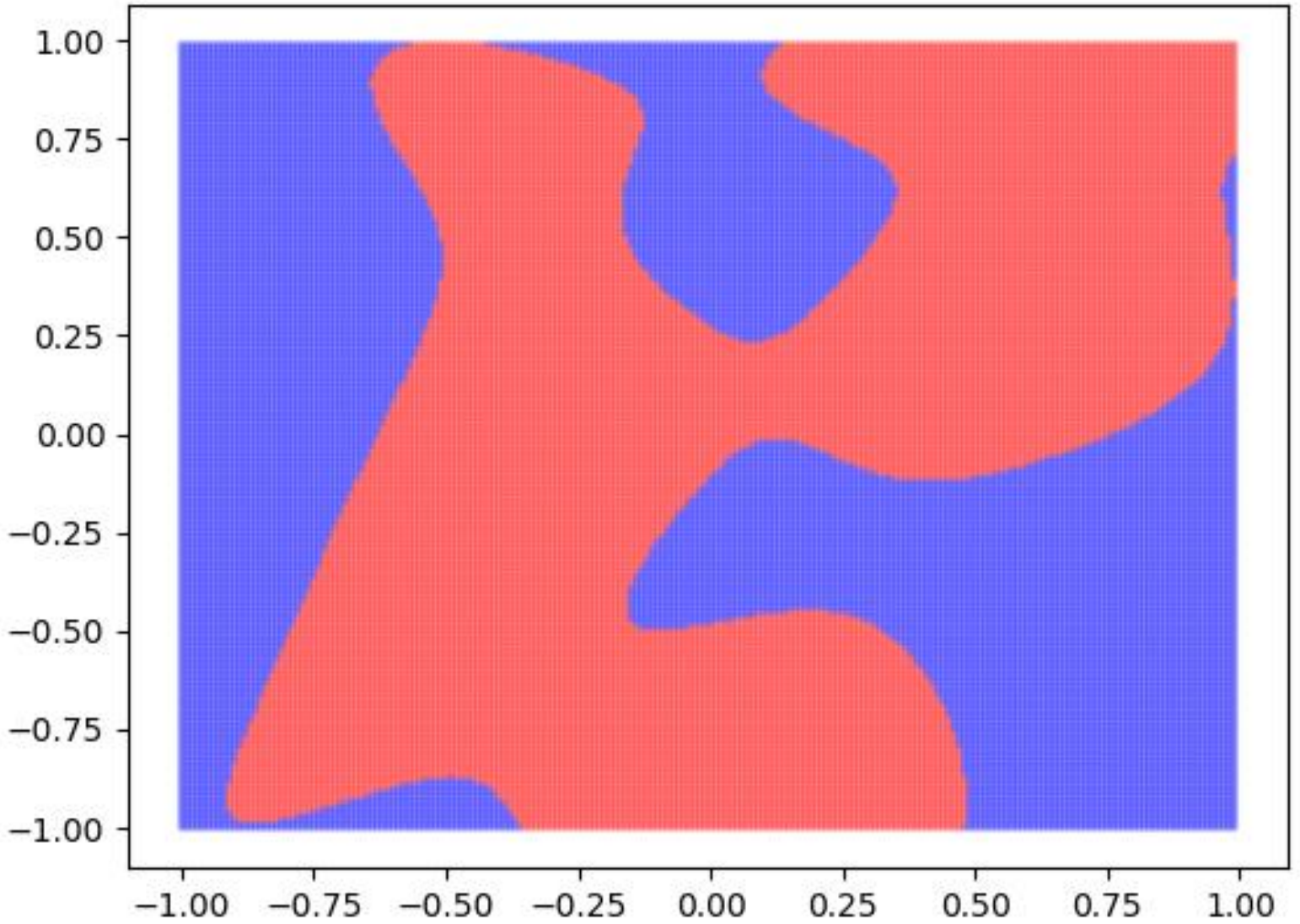

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

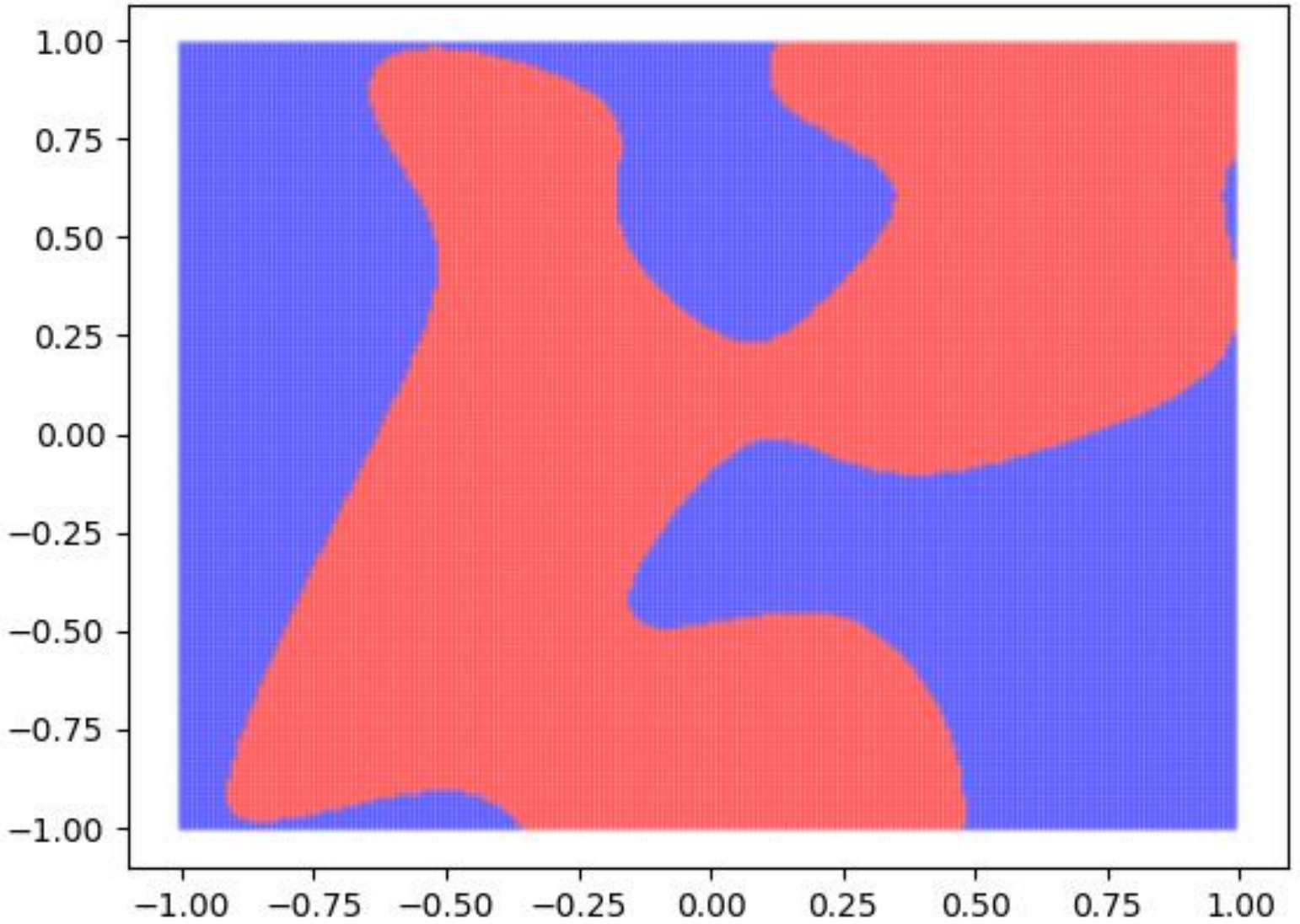

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

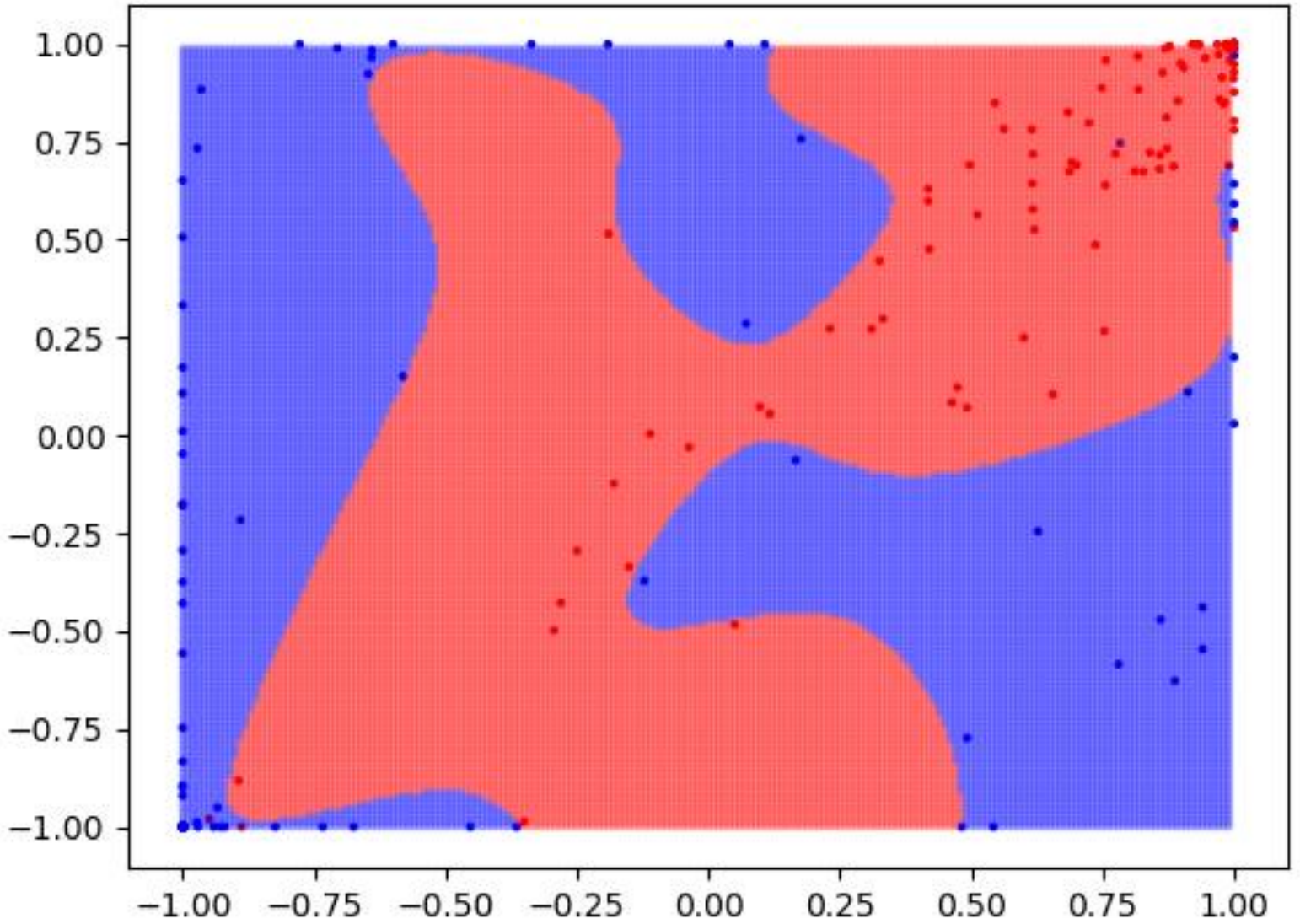

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

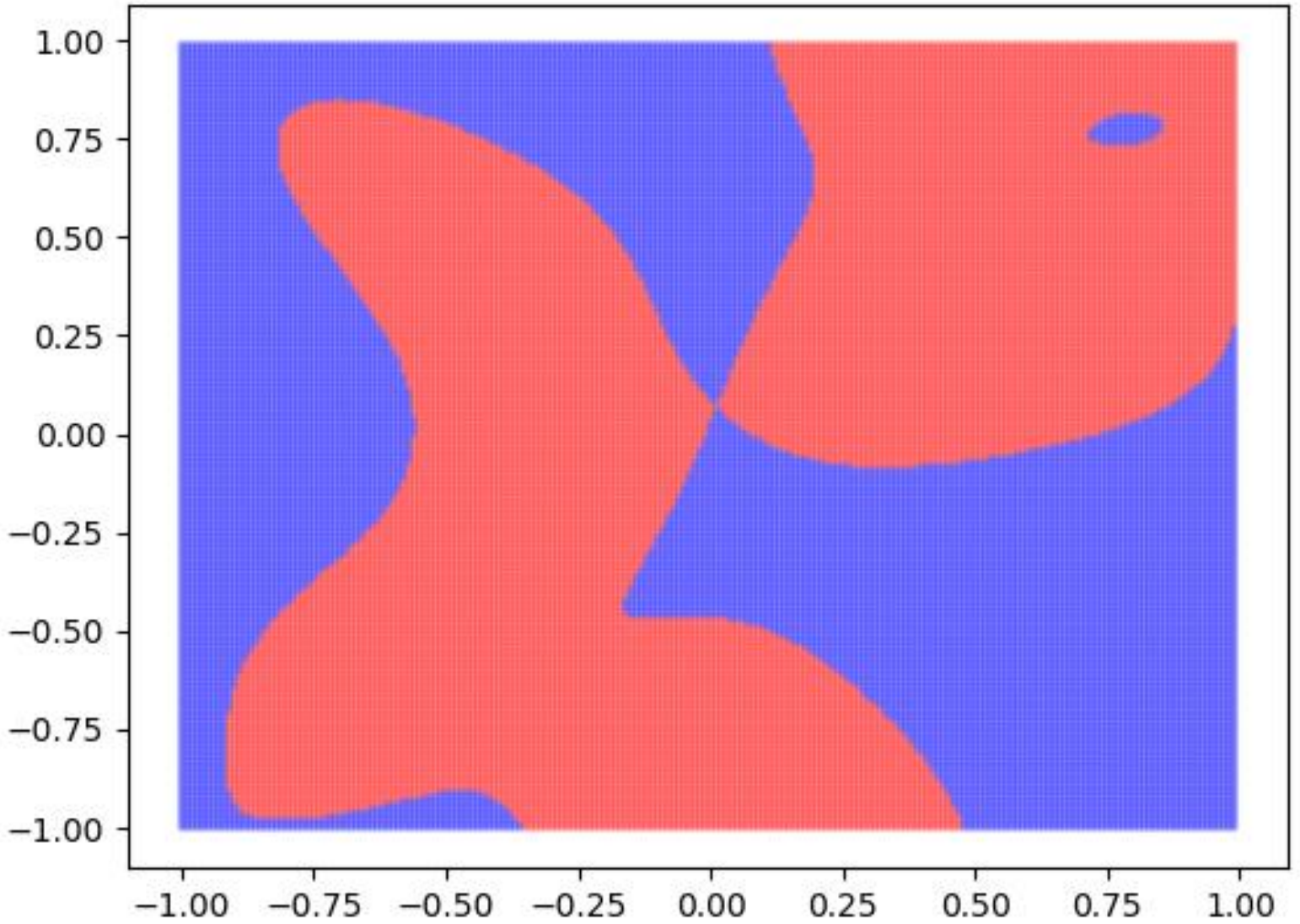

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

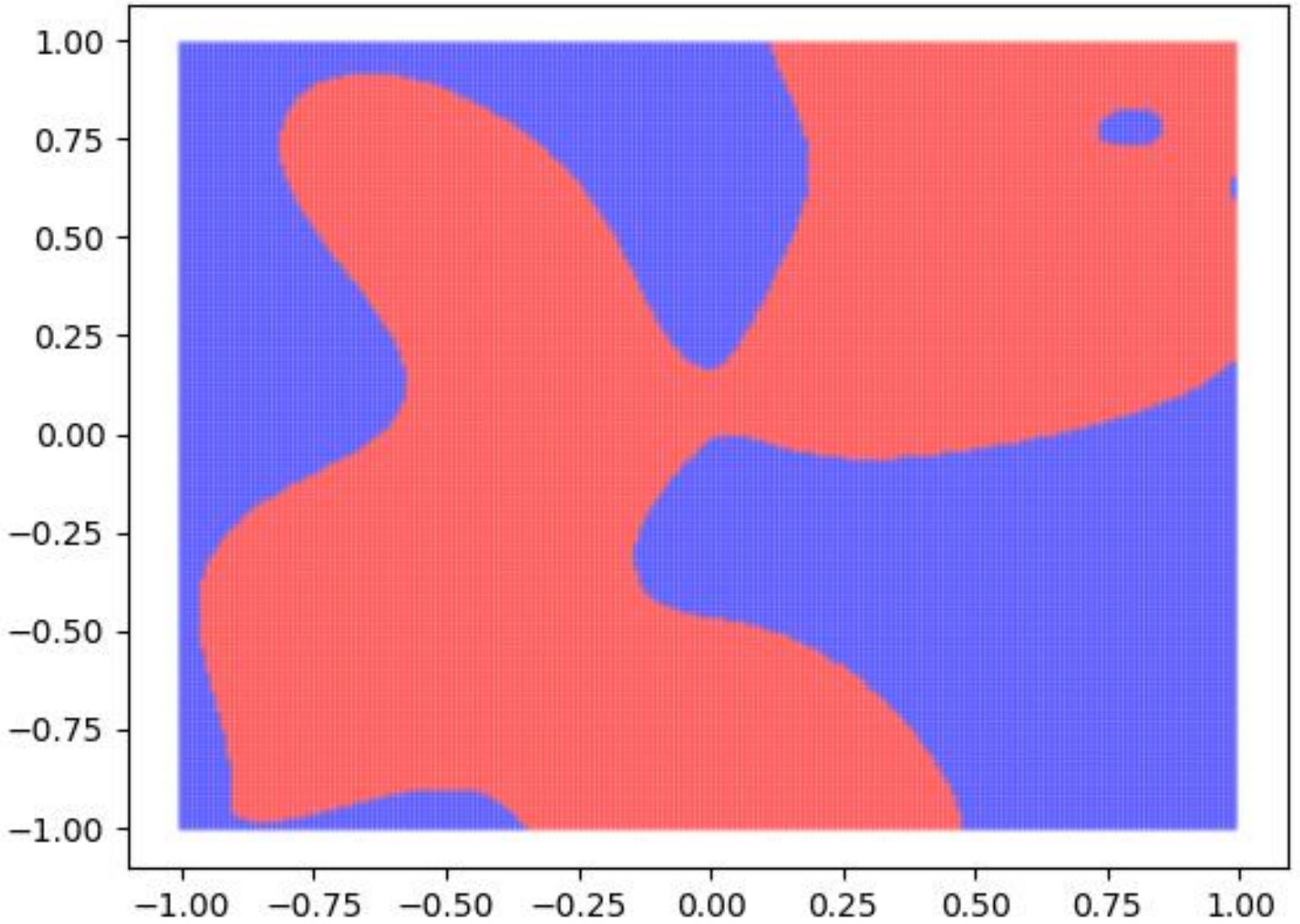

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

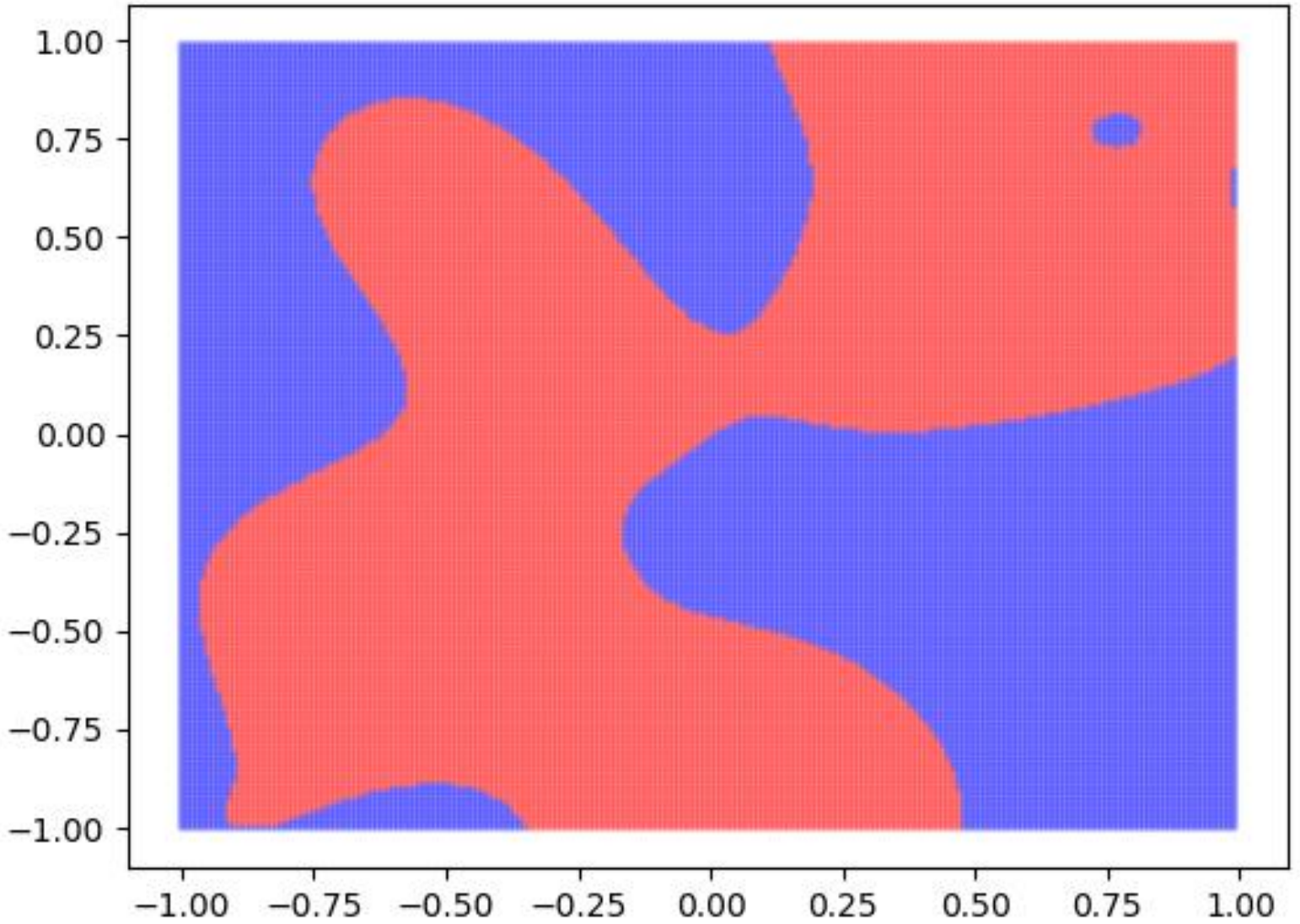

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

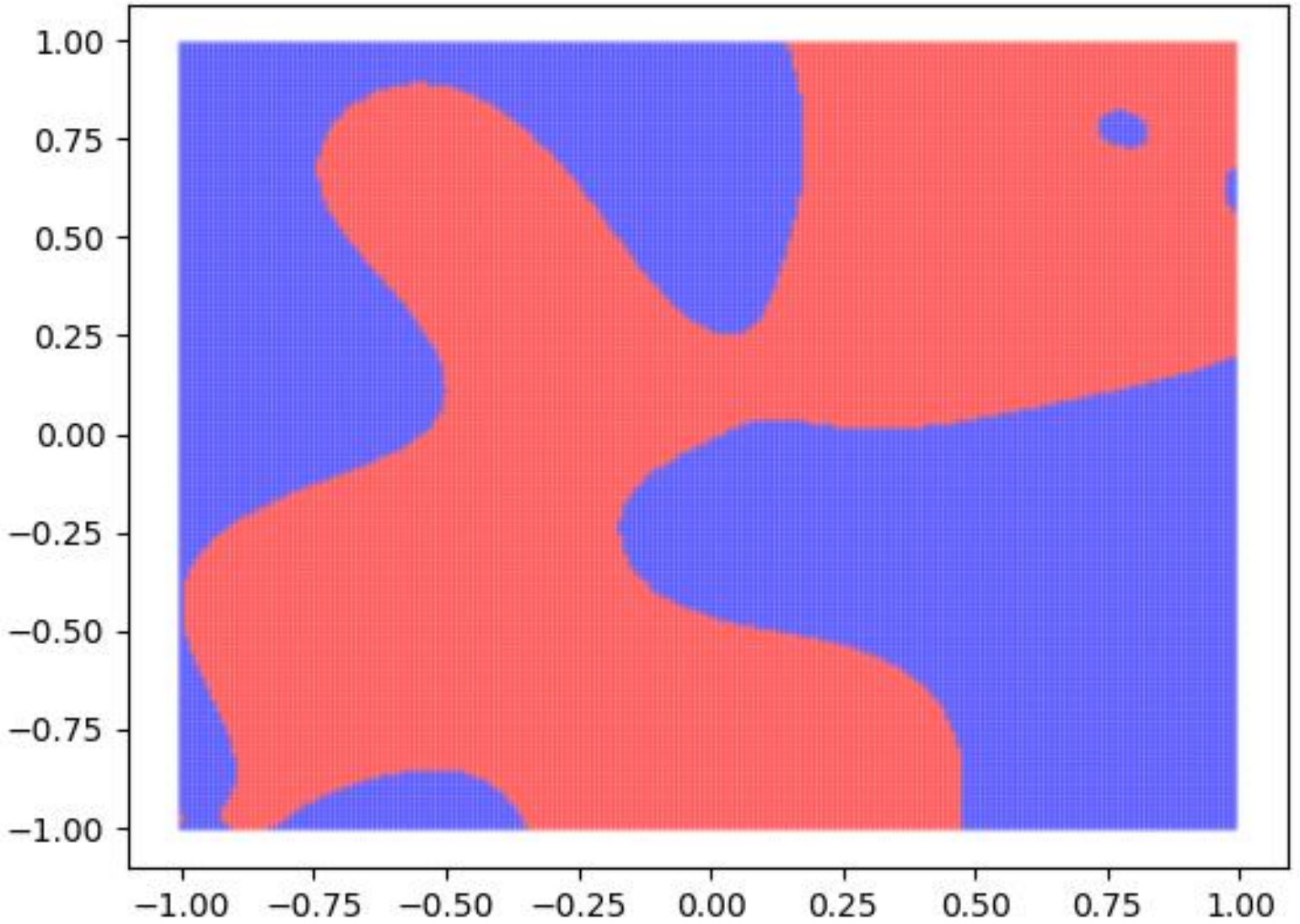

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

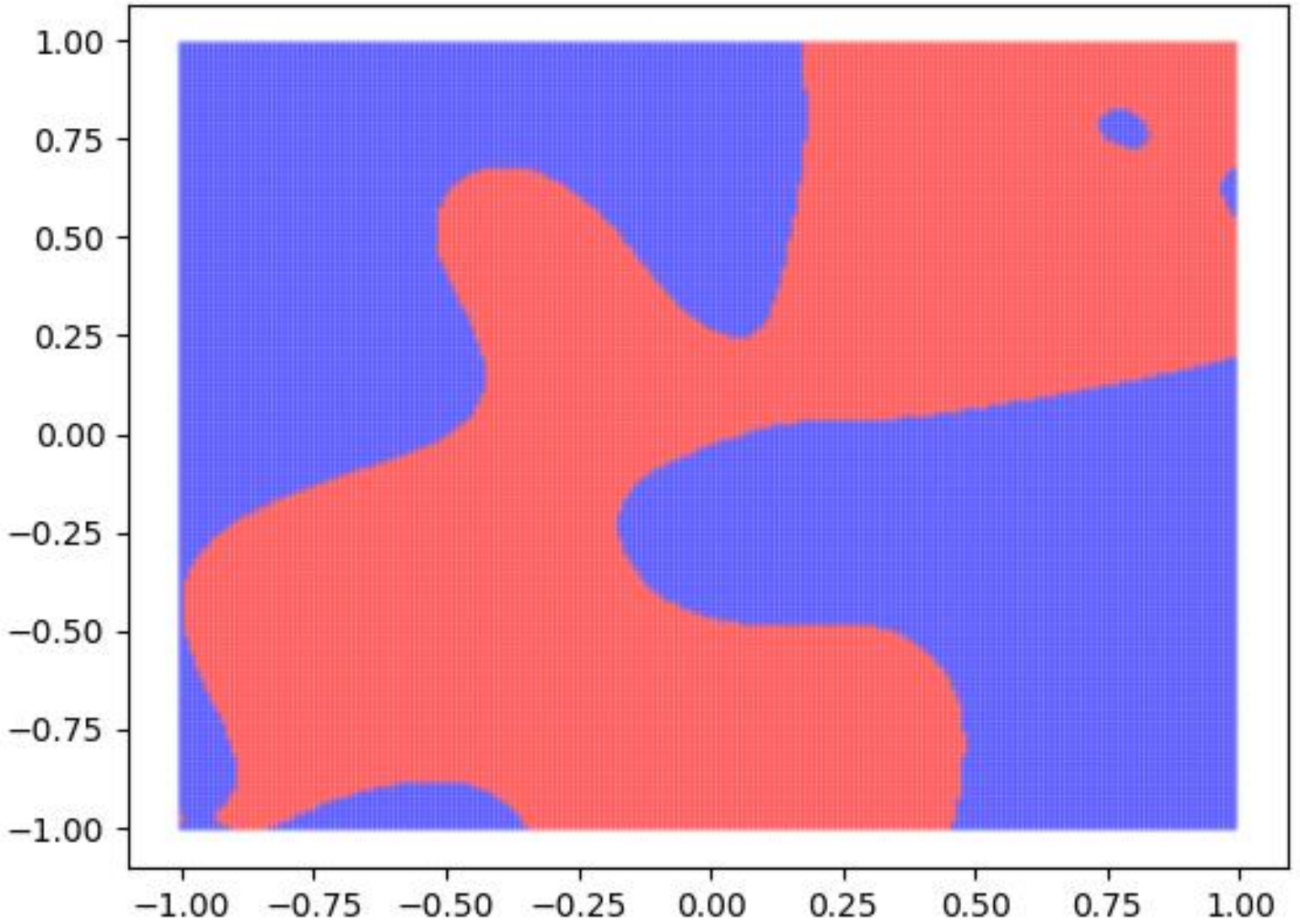

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

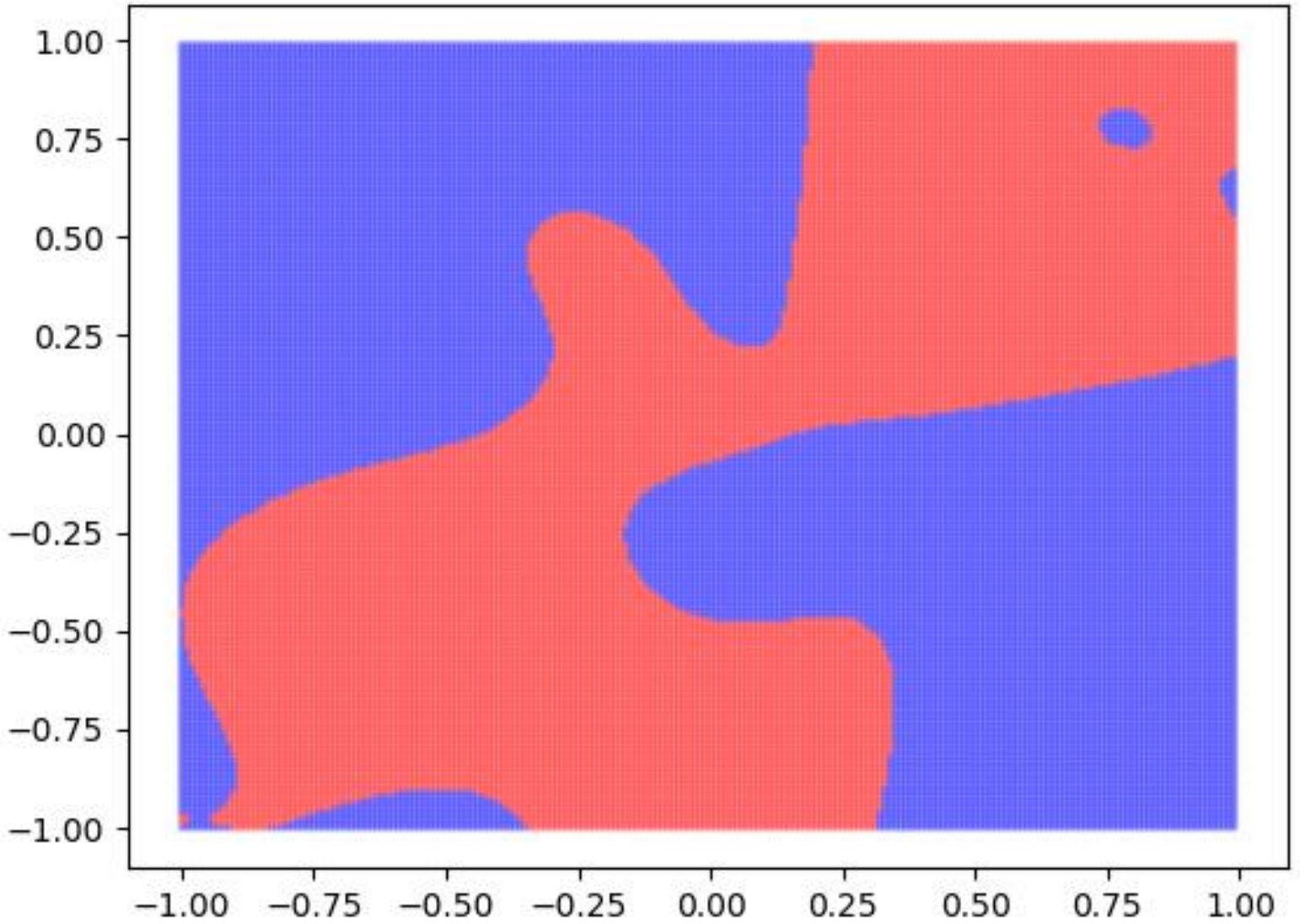

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

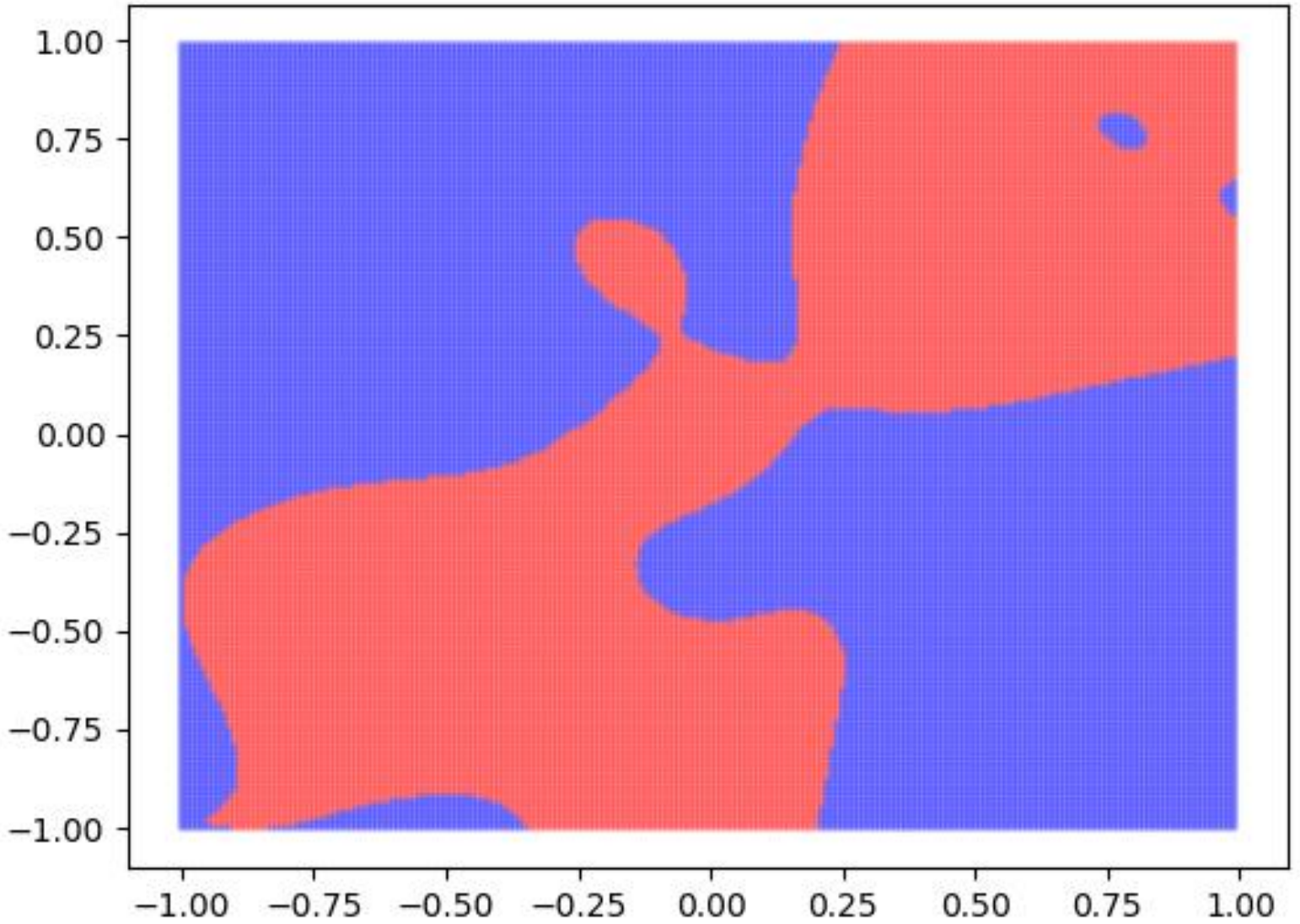

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

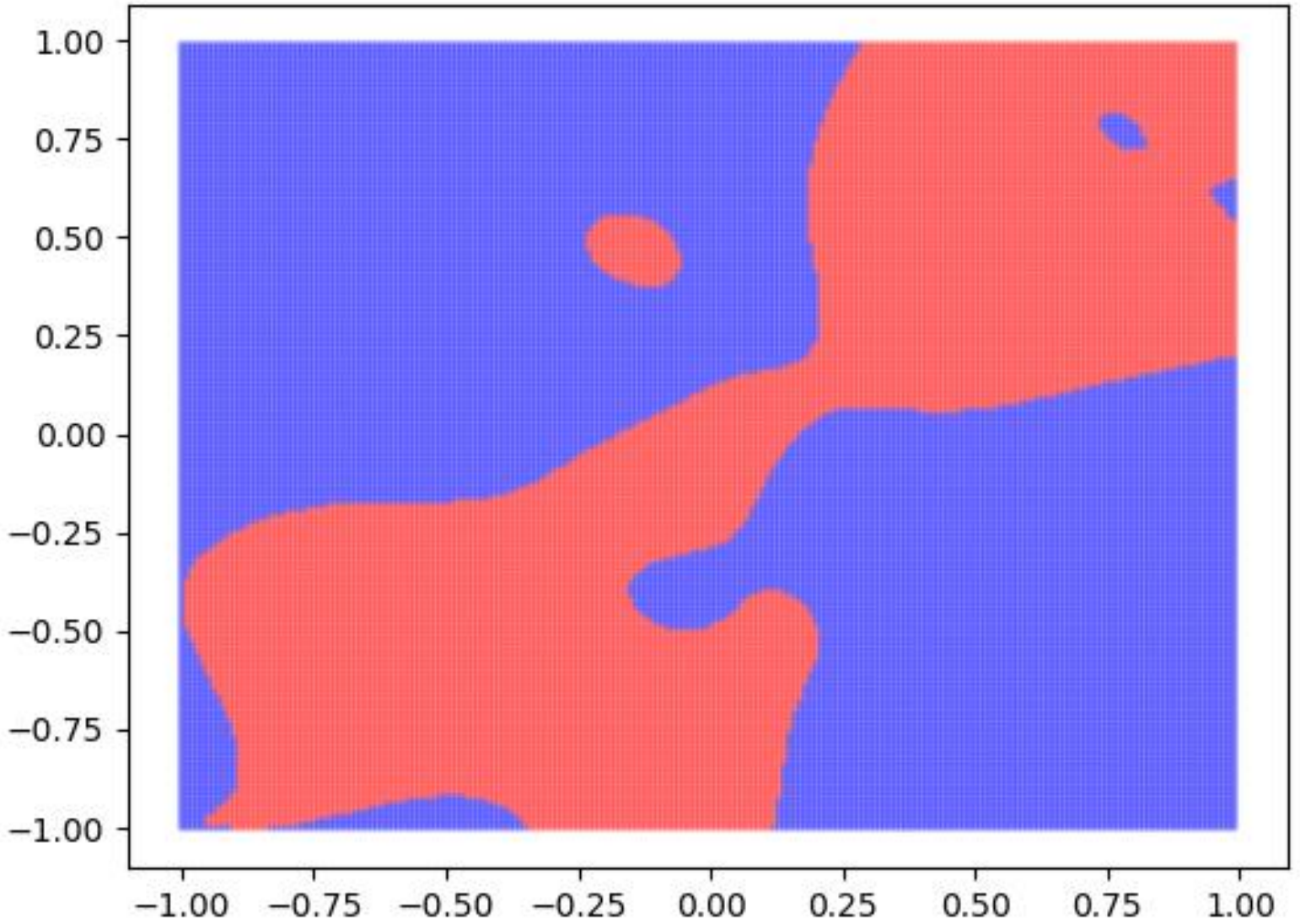

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

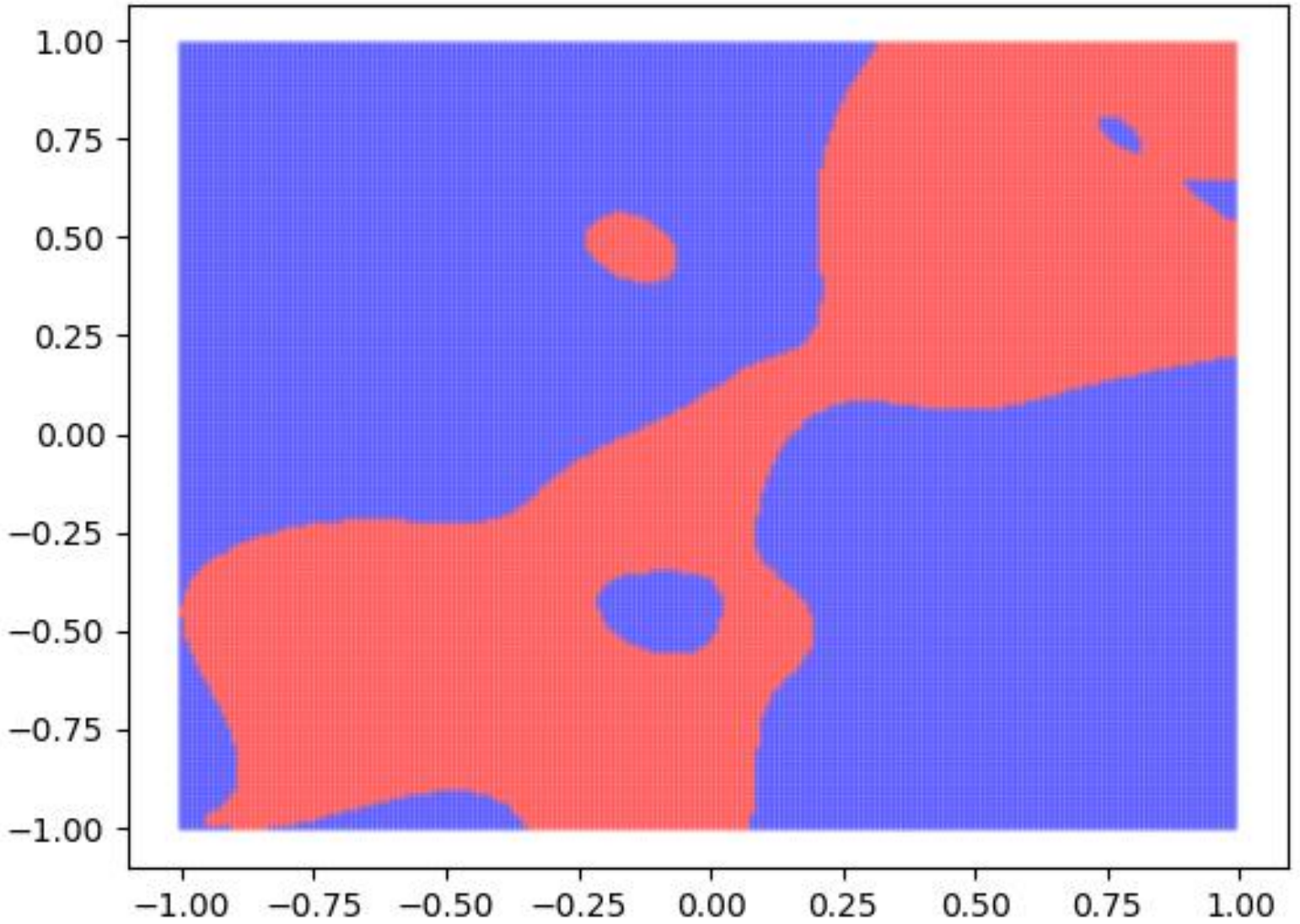

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

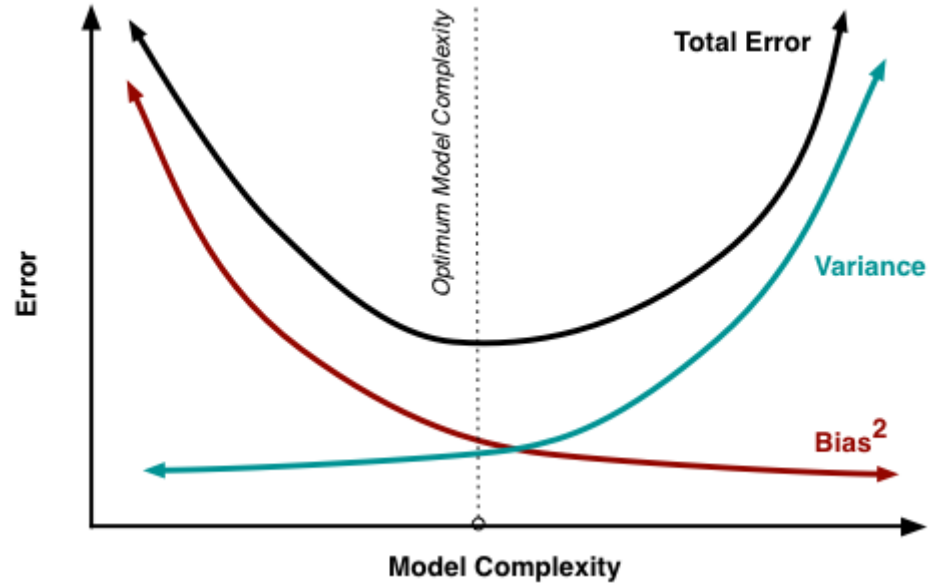

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

Over/underfitting

We also now have a concept of "training error"

- What is the training error for the SVM?

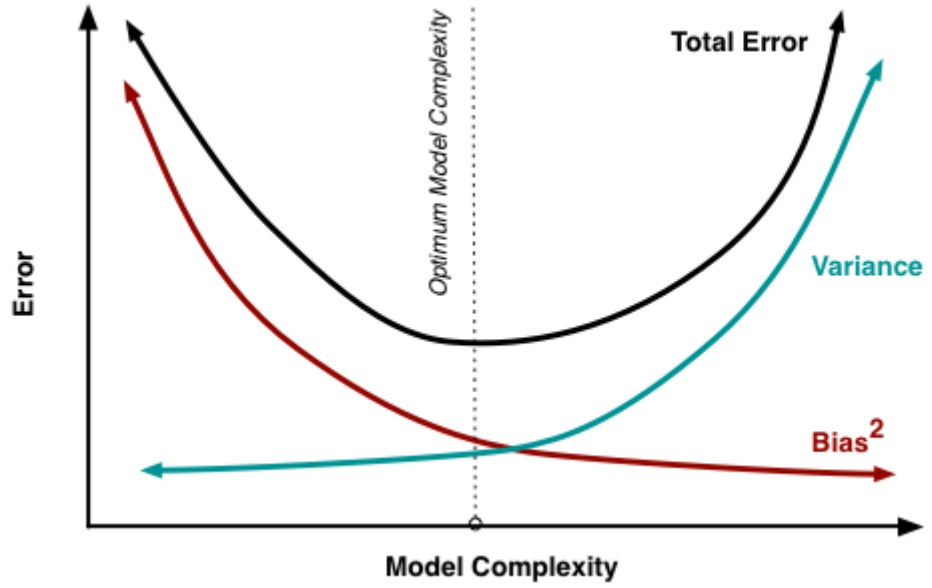
$$\sum \xi_i$$



Over/underfitting

We also now have a concept of "training error"

- What is the training error for the SVM?
- As complexity increases, should the error go up or down?

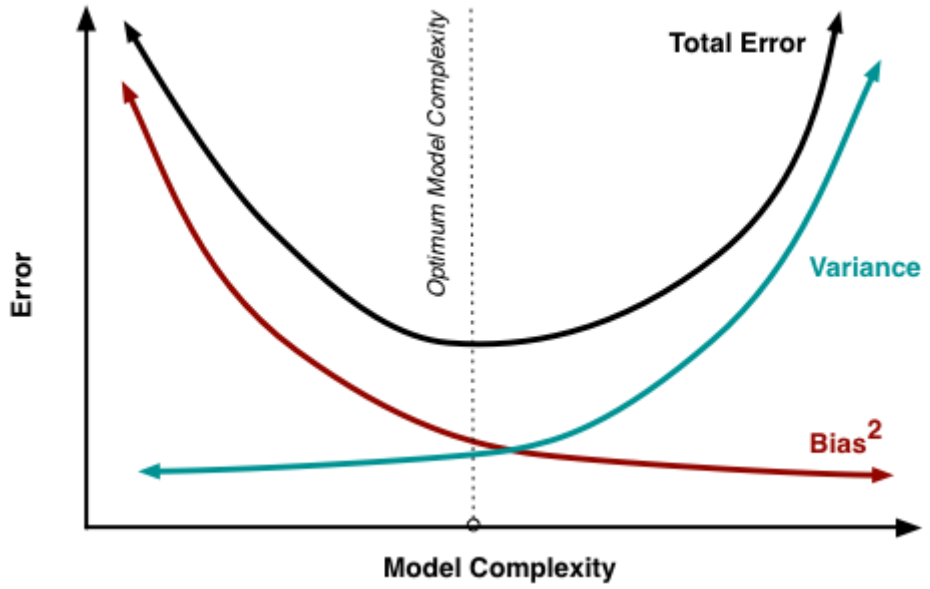


Over/underfitting

We also now have a concept of "training error"

- What is the training error for the SVM?
- As complexity increases, should the error go up or down?
- Do we have a variance for this error?

Variance of the 10-folds



Over/underfitting

We're now also considering a much larger set of models

- kNN, we only did 25 models
- SVM, we could do thousands of them

We pay a penalty when we select the "best" model from a large set of hypothesis models, but we may want to try several in order to be certain

Next, we'll start Neural Networks and this will be the focus of the next couple weeks

- Even more complexity
- Even less interpretability
- Very good at predicting, however