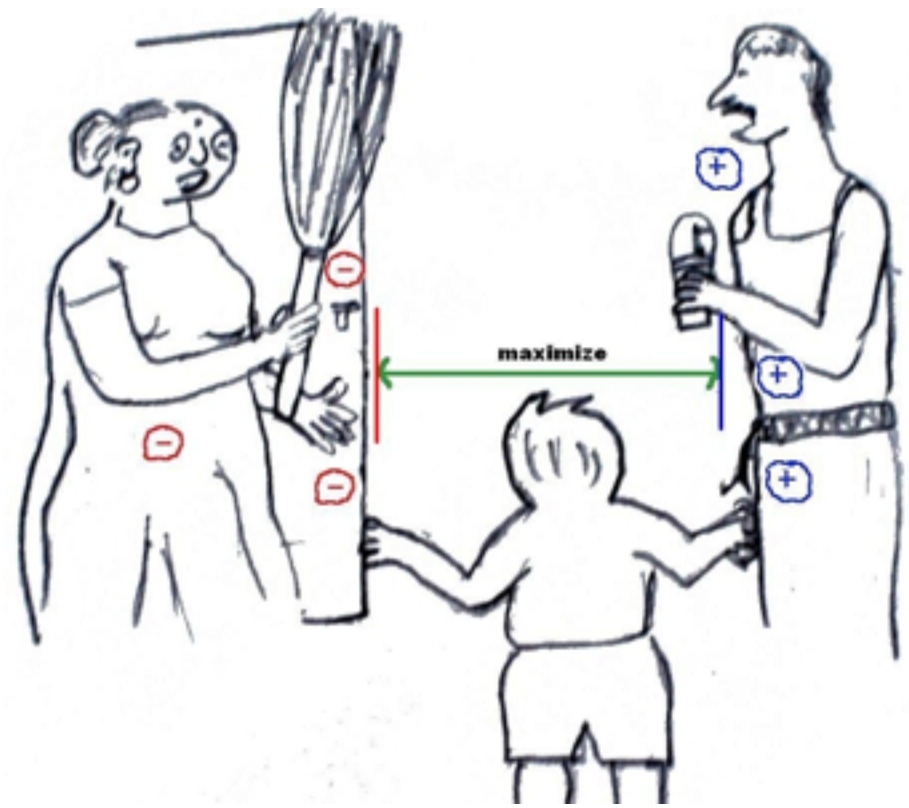
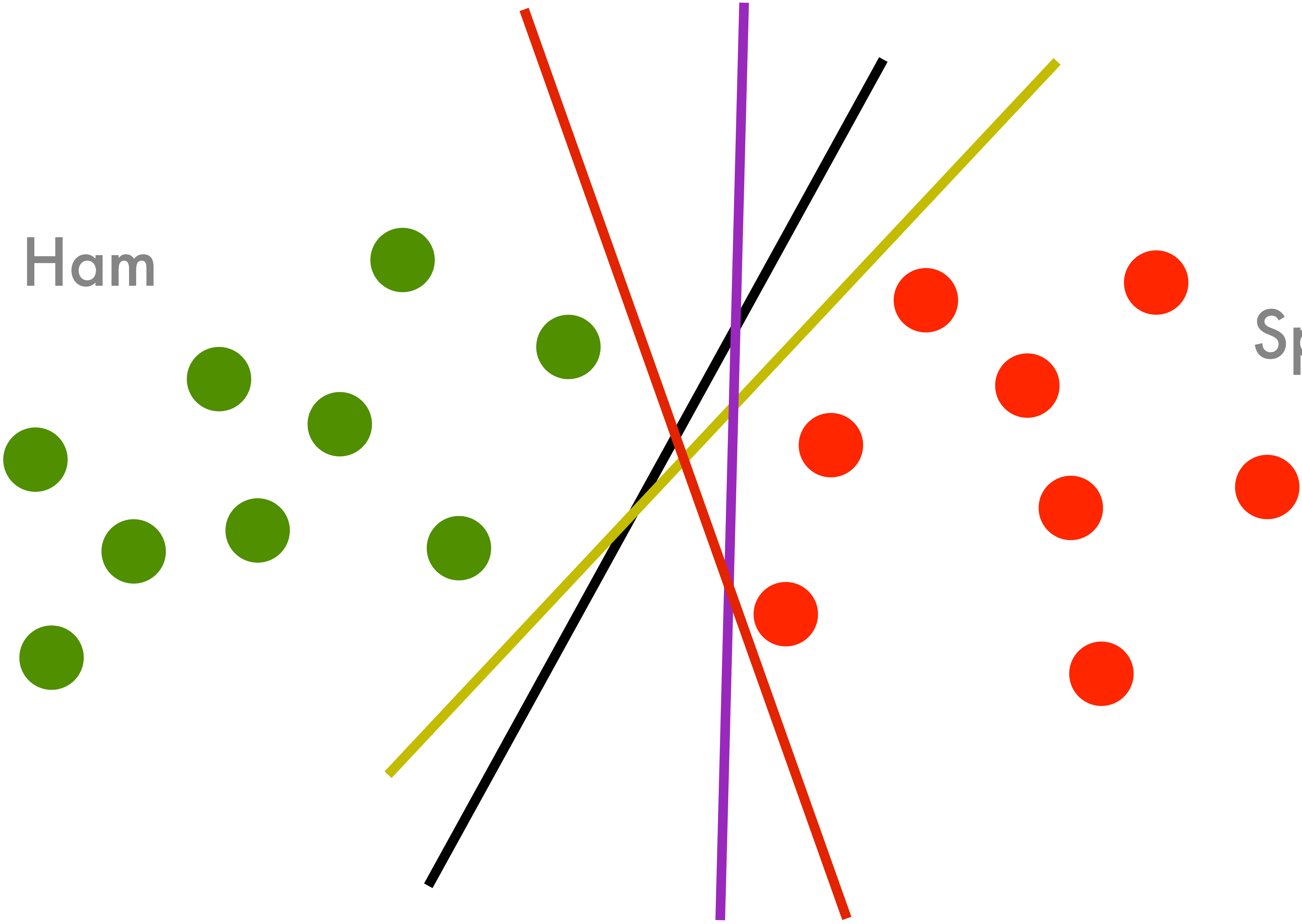
Support Vector Machines

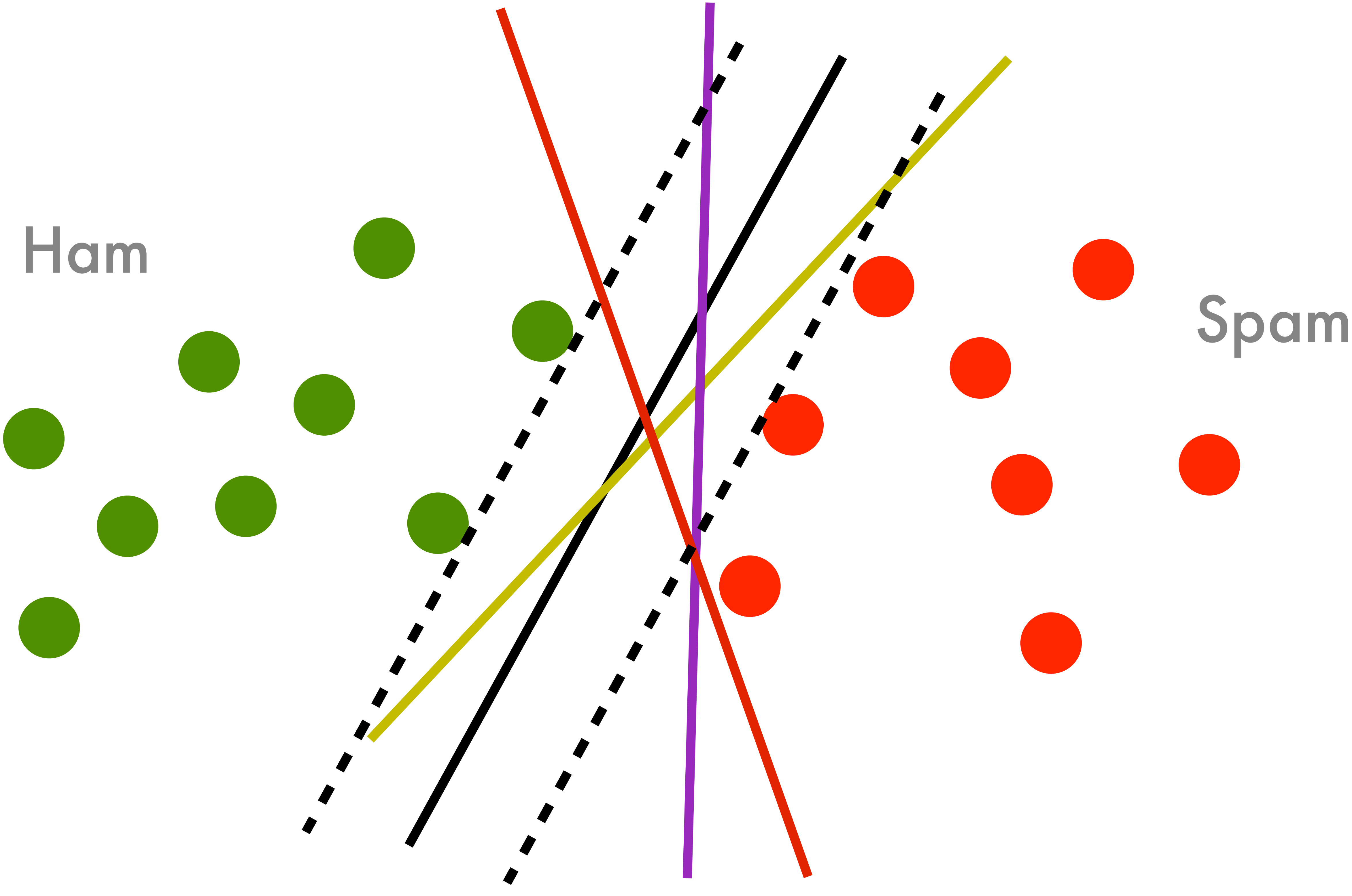


http://maktoons.blogspot.com/2009/03/support-vector-machine.html

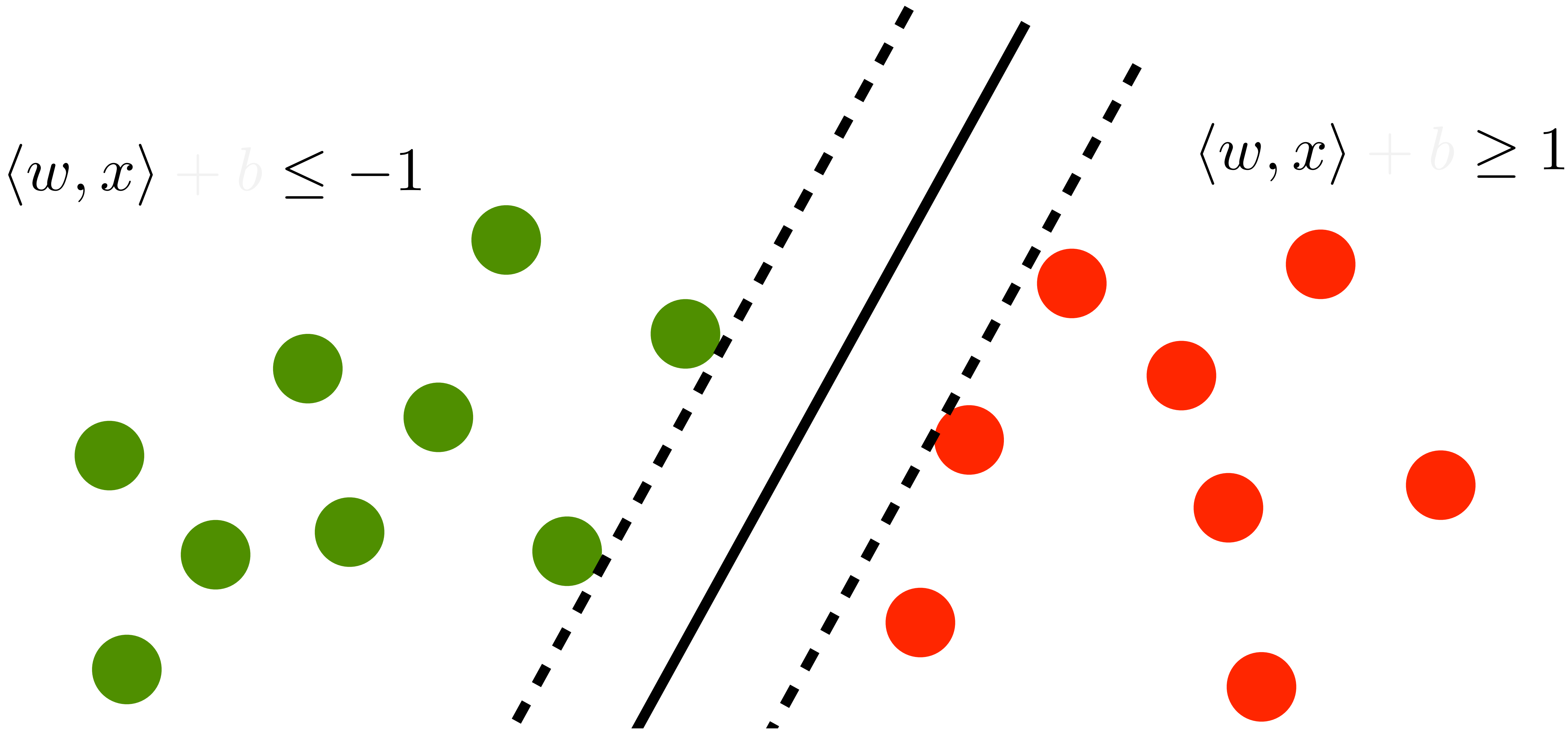
Linear Separator



Linear Separator



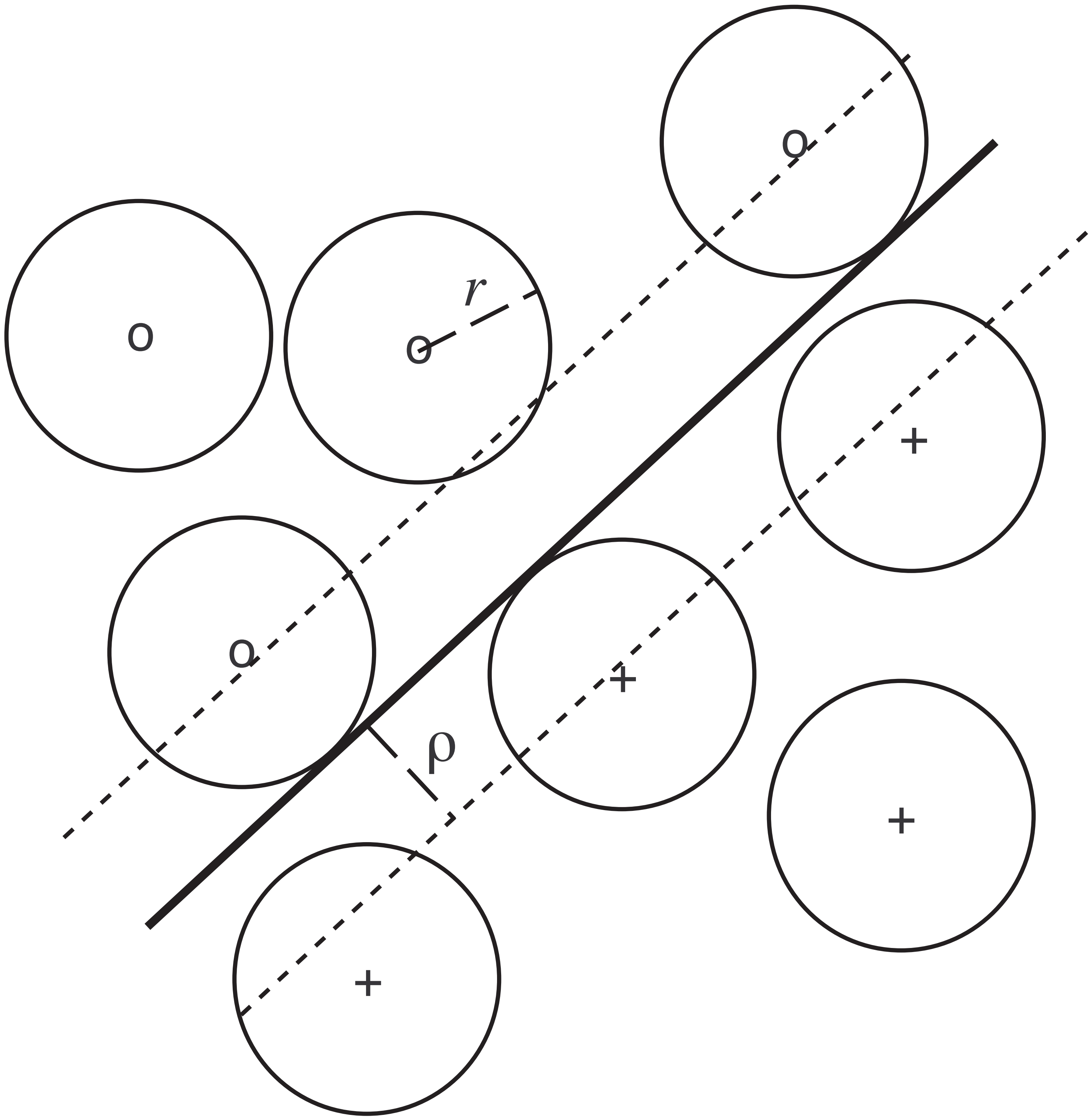
Large Margin Classifier



linear function

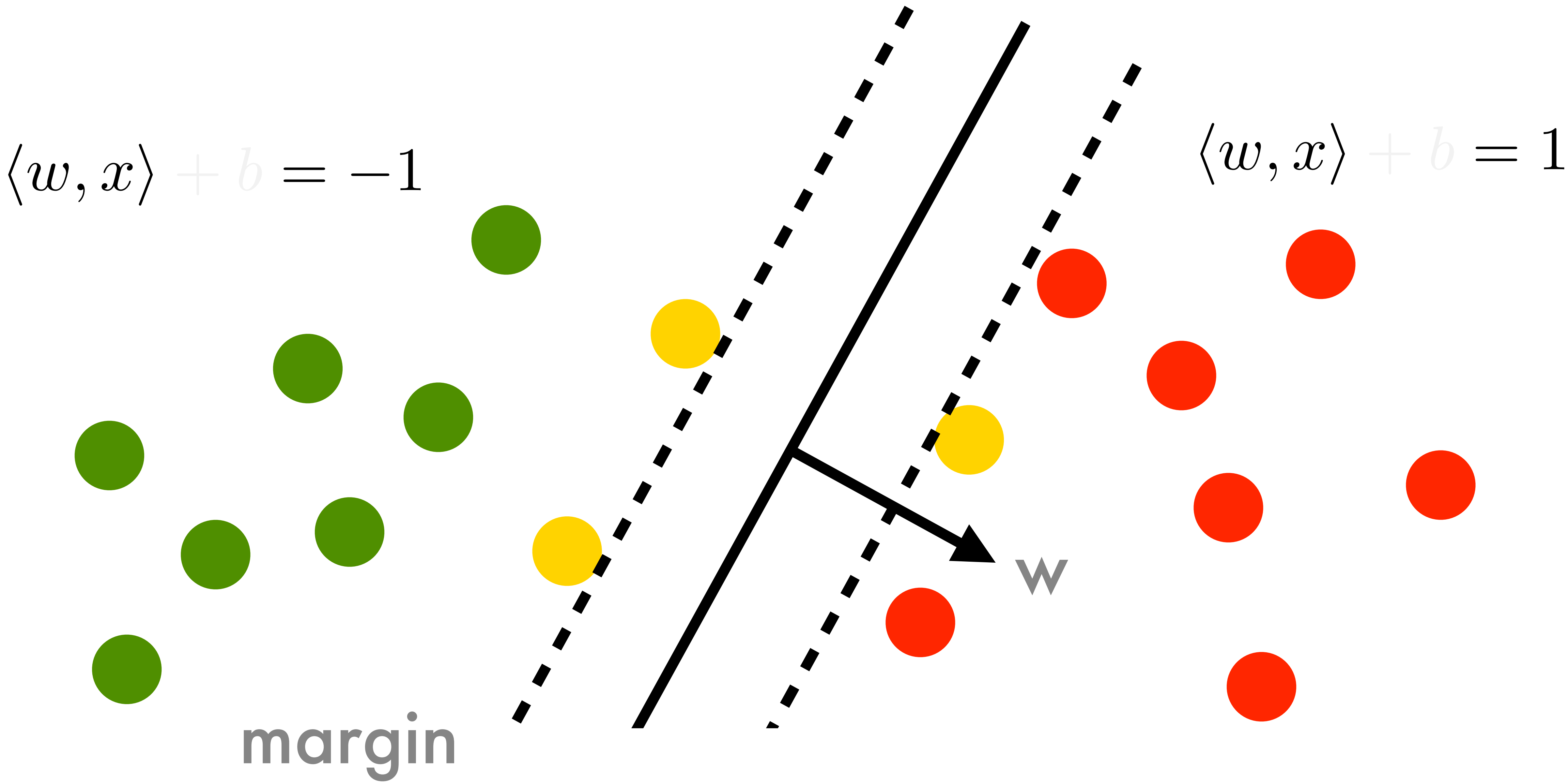
$$f(x) = \langle w, x \rangle + b$$

Why large margins?



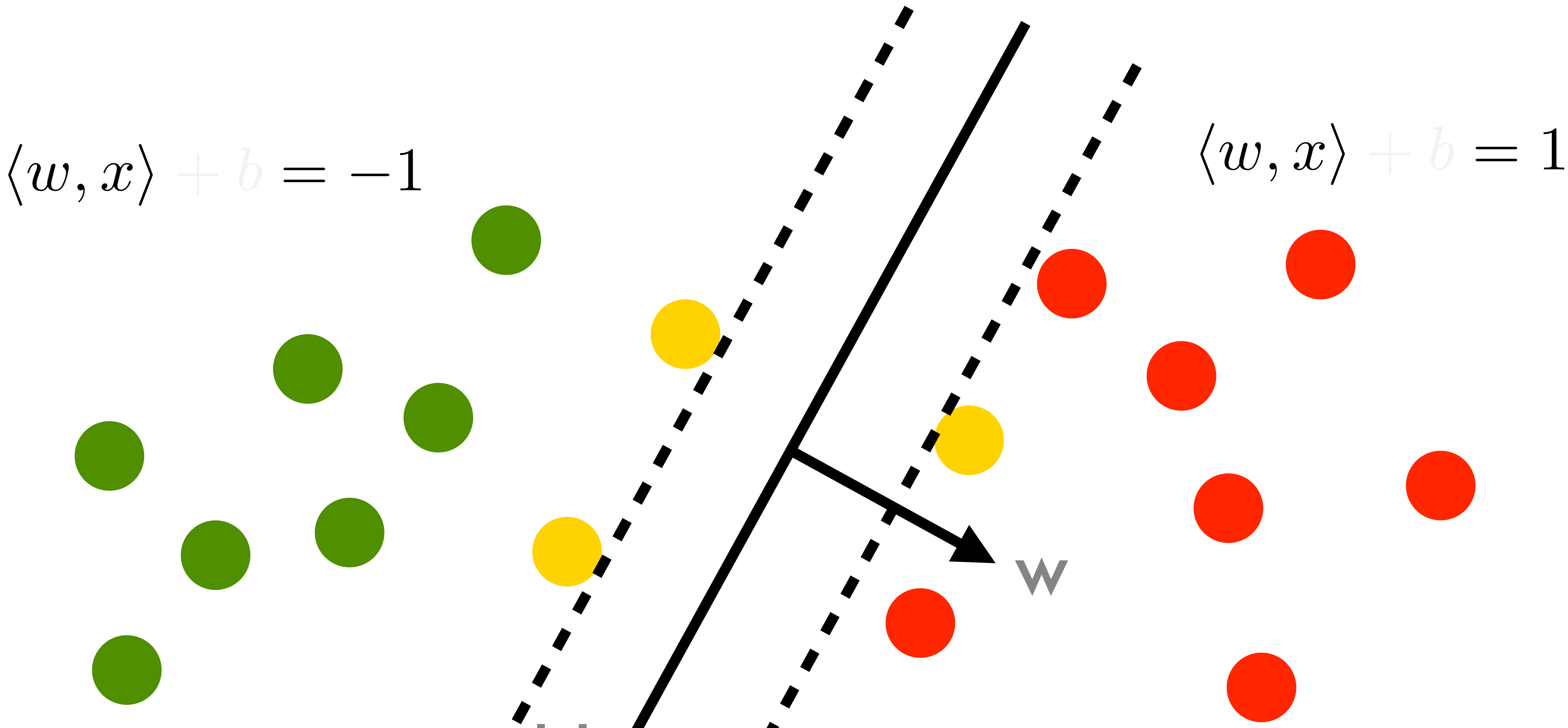
- Maximum robustness relative to uncertainty
- Symmetry breaking
- Independent of correctly classified instances
- Easy to find for easy problems

Large Margin Classifier



$$\frac{\langle x_+ - x_-, w \rangle}{2\|w\|} = \frac{1}{2\|w\|}\left[\left[\langle x_+, w \rangle + b\right] - \left[\langle x_-, w \rangle + b\right]\right] = \frac{1}{\|w\|}$$

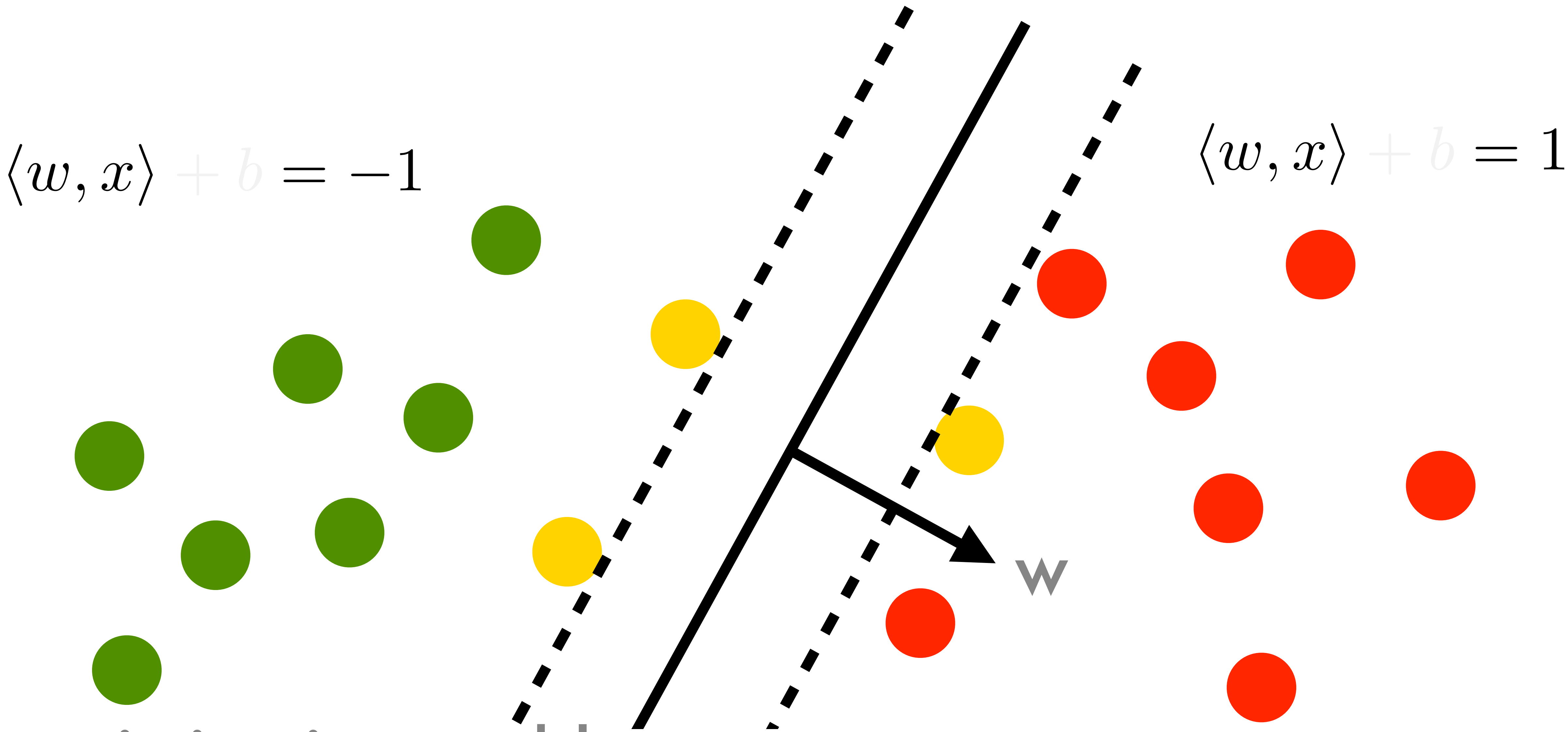
Large Margin Classifier



optimization problem

$$\underset{w,b}{\text{maximize}} \frac{1}{\|w\|} \text{ subject to } y_i \left[\langle x_i, w \rangle + b \right] \geq 1$$

Large Margin Classifier



optimization problem

$$\underset{w,b}{\text{minimize}} \; \frac{1}{2} \|w\|^2 \text{ subject to } y_i \left[\langle x_i, w \rangle + b \right] \geq 1$$

Lagrangian

- **Primal optimization problem**

$$\underset{w}{\operatorname{minimize}} \frac{1}{2}\|w\|^2 \text{ subject to } y_i \langle x_i, w\rangle \geq 1$$

- **Lagrange function**

constraint

$$L(w, \alpha) = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i \left[y_i \langle x_i, w\rangle - 1\right]$$

Derivatives in w need to vanish

$$\partial_w L(w, a) = w - \sum_i \alpha_i y_i x_i = 0$$

$$w = \sum_i y_i \alpha_i x_i$$

Geometry of Lagrangian

Constrained Optimization

constraint

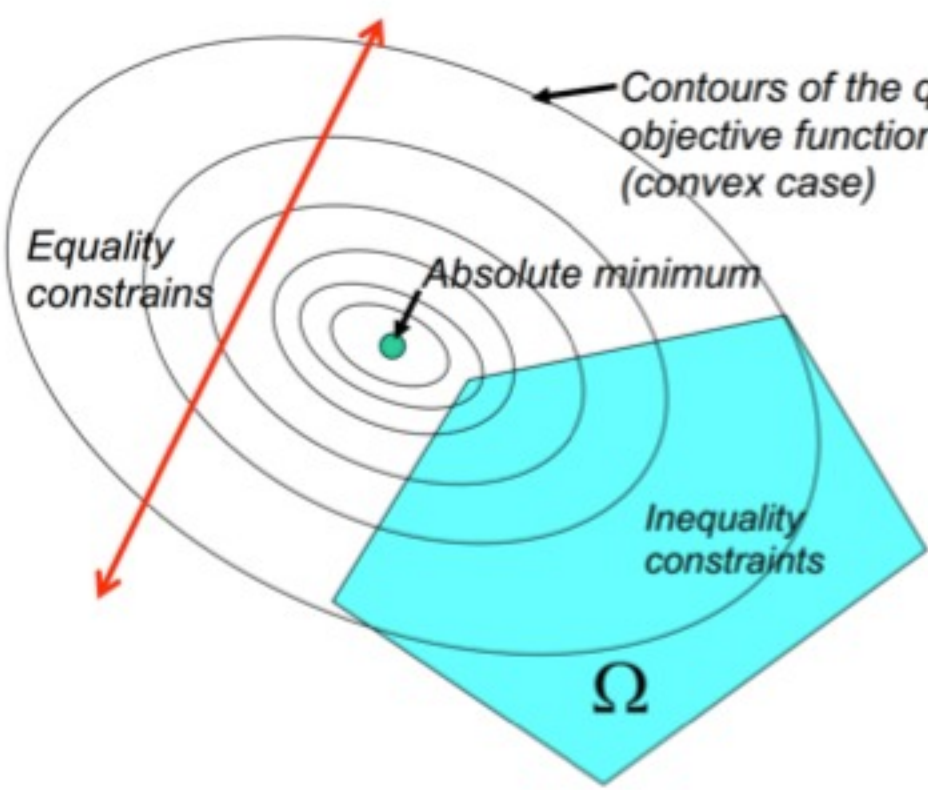
$$\underset{w,b}{\text{minimize}} \ \frac{1}{2}\|w\|^2 \ \text{subject to} \ y_i \left[\langle x_i, w\rangle + b\right] \geq 1$$

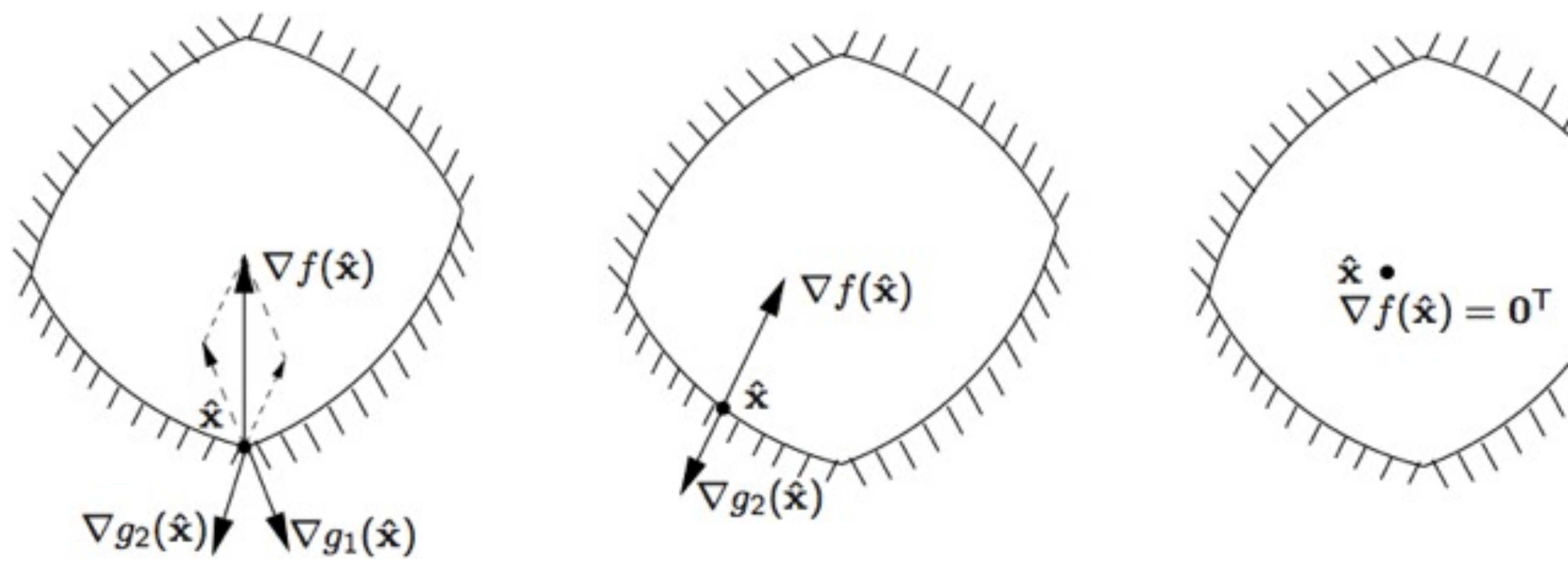
$$w = \sum_i y_i \alpha_i x_i$$

- Quadratic Programming
  - Quadratic Objective
  - Linear Constraints

KKT condition: optimal point is achieved at *active constraints* where $a_i > 0$ ($a_i = 0$ => inactive)

$$\alpha_i \left[y_i \left[\langle w, x_i\rangle + b\right] - 1\right] = 0$$



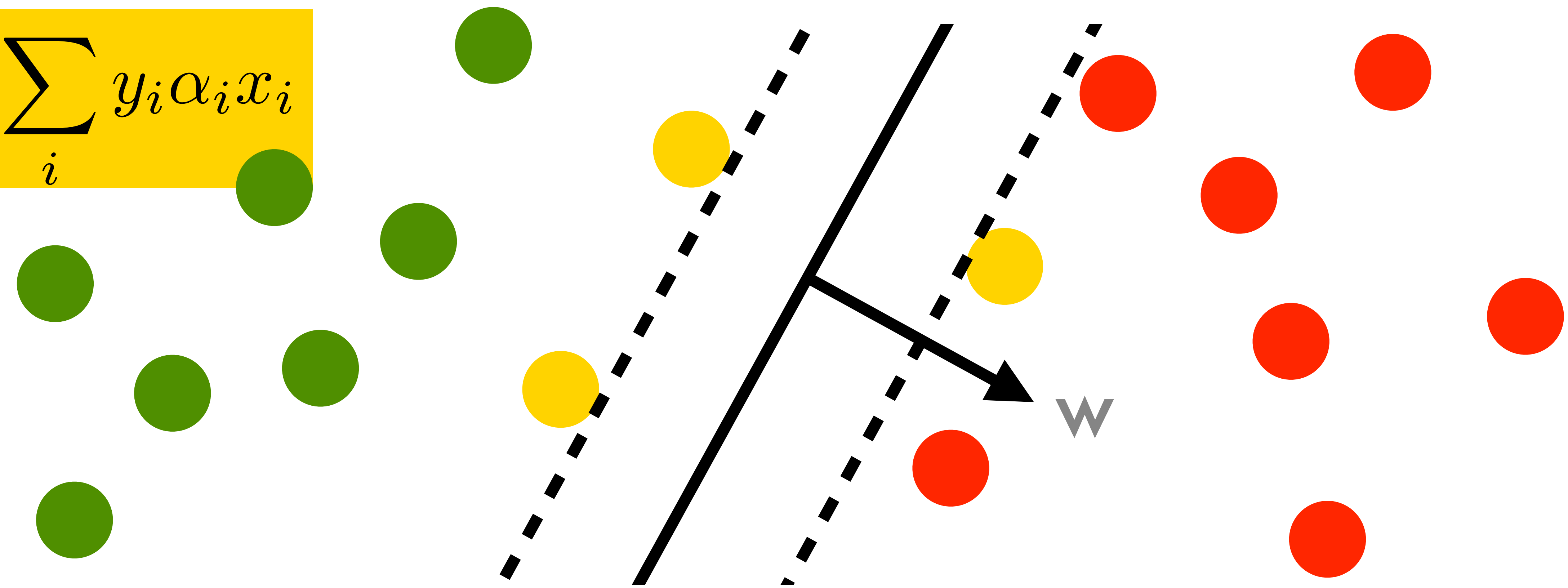


Geometry of KKT

KKT => Support Vectors

$$\underset{w,b}{\text{minimize}} \frac{1}{2} \|w\|^2 \text{ subject to } y_i \left[\langle x_i, w \rangle + b\right] \geq 1$$

$$w = \sum_i y_i \alpha_i x_i$$



Karush Kuhn Tucker (KKT)
Optimality Condition

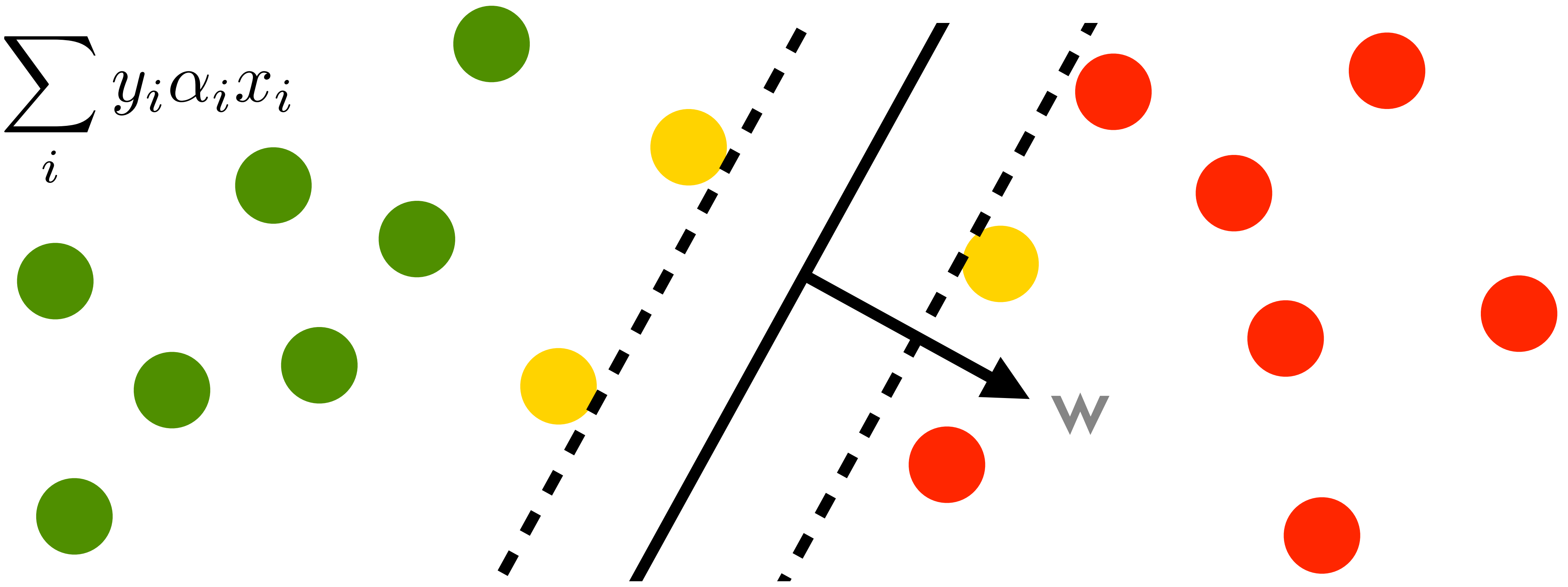
$$\alpha_i \left[y_i \left[\langle w, x_i \rangle + b\right] - 1\right] = 0$$

$$\alpha_i = 0$$

$$\alpha_i > 0 \implies y_i \left[\langle w, x_i \rangle + b\right] = 1$$

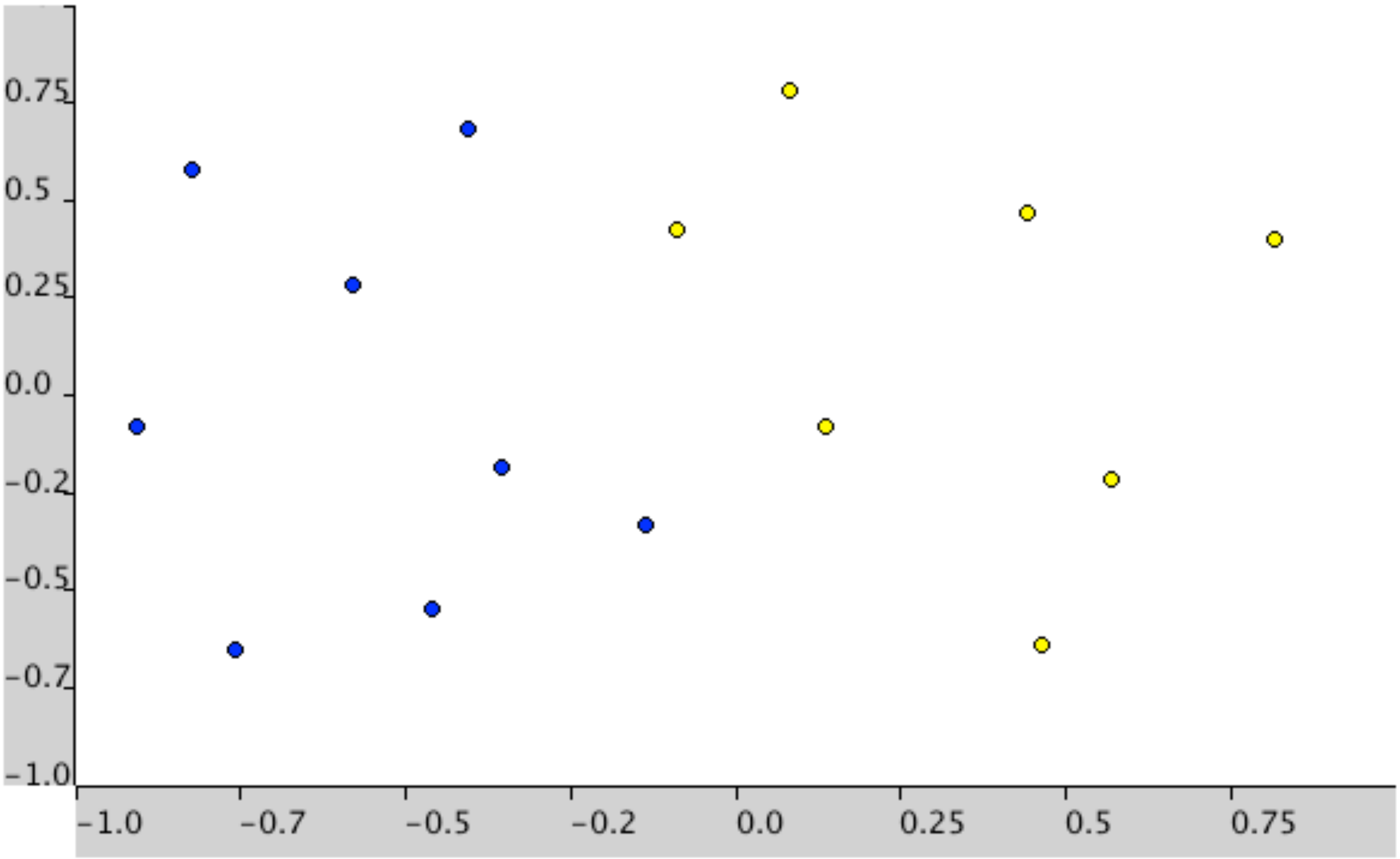
Properties

$$w = \sum_i y_i \alpha_i x_i$$



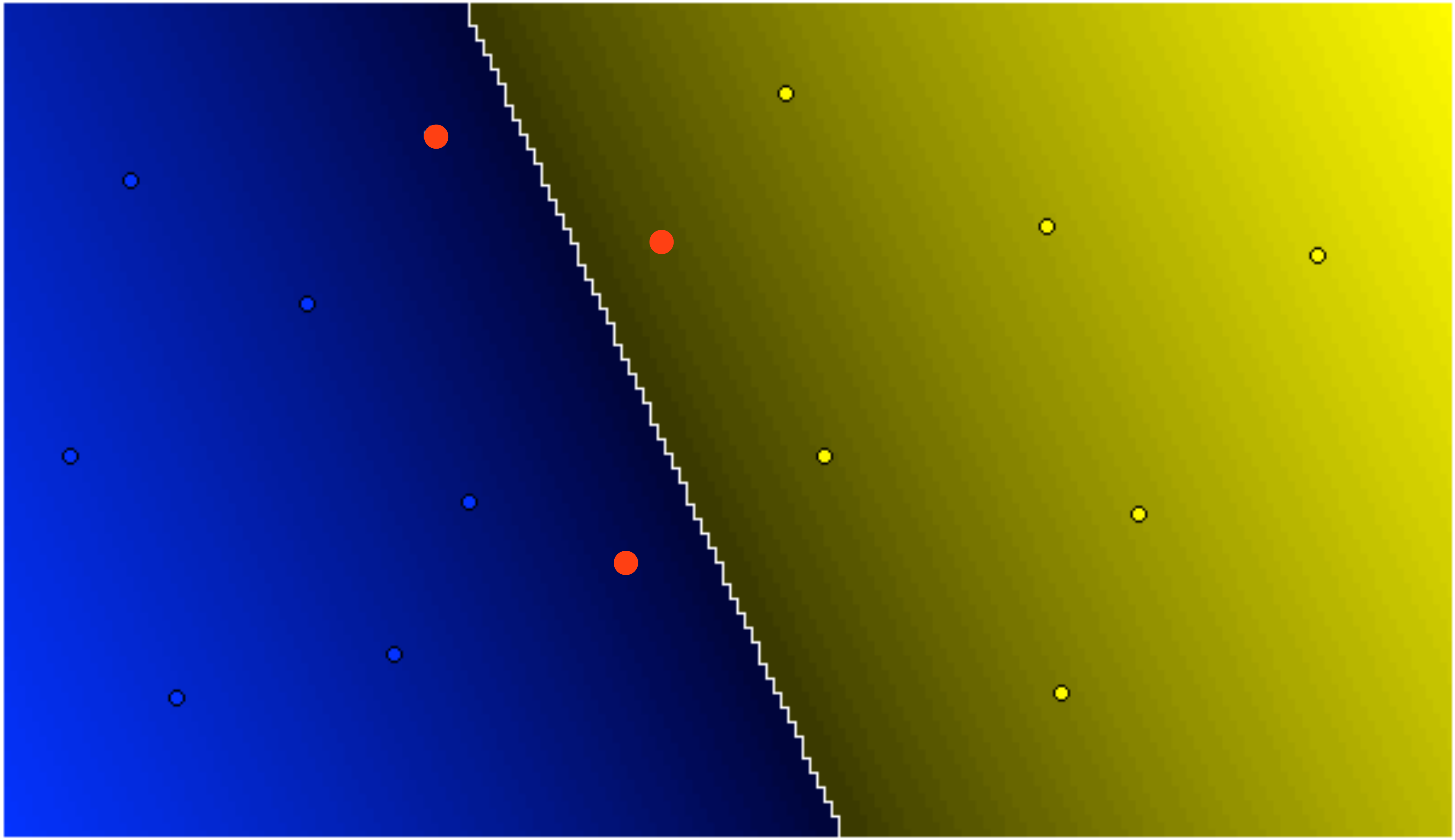
- Weight vector w as weighted linear combination of instances
- Only points on margin matter (ignore the rest and get same solution)
- Only inner products matter
  - Quadratic program
  - We can replace the inner product by a kernel
- Keeps instances away from the margin

Example



Example

Number of Support Vectors: **3** (-ve: 2, +ve: 1) Total number of points: 15



Alternative: Dual Problem

- Lagrange function

$$L(w, \alpha) = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i \left[y_i \langle x_i, w \rangle - 1 \right]$$

- Derivatives in w need to vanish

$$\partial_w L(w, a) = w - \sum_i \alpha_i y_i x_i = 0$$

$$w = \sum_i y_i \alpha_i x_i$$

- Plugging w back into L yields

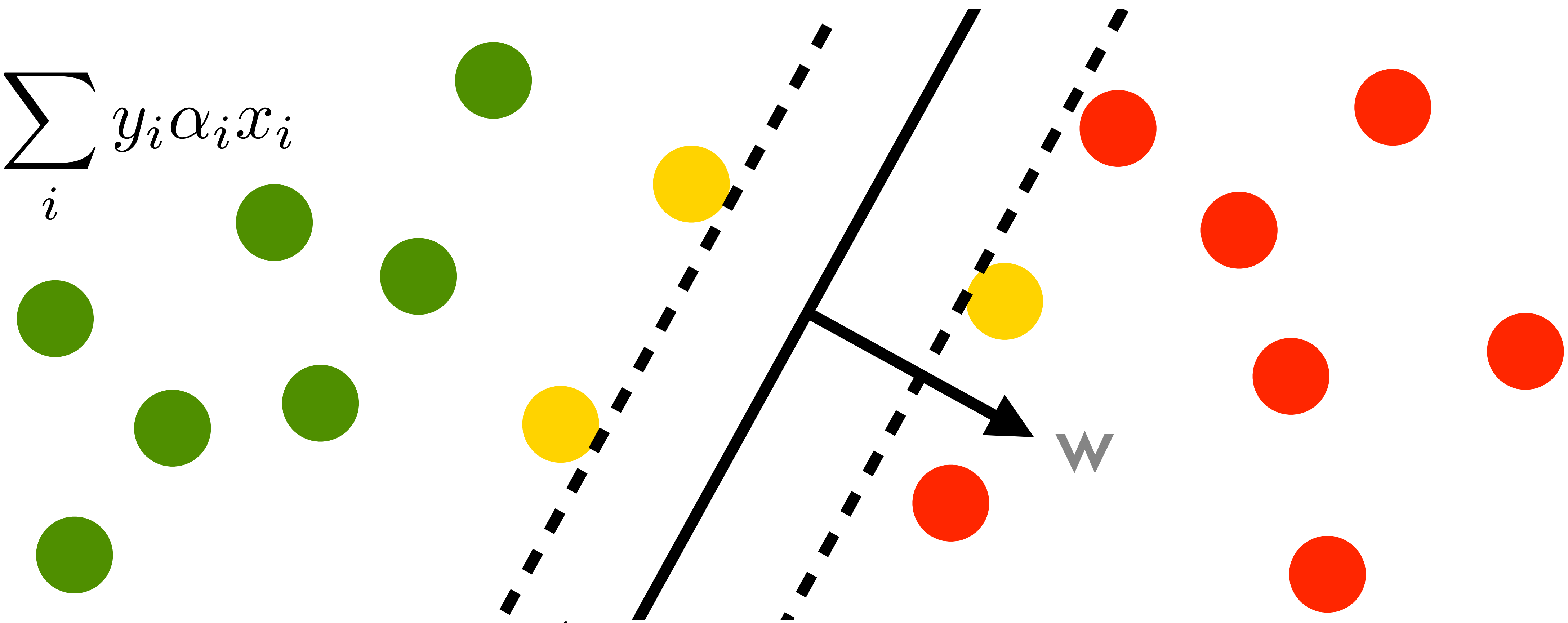
$$\underset{\alpha}{\text{maximize}} - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i$$

subject to **Lagrangian** $\alpha_i \geq 0$

Primal vs. Dual

Primal $\underset{w,b}{\text{minimize}}\ \frac{1}{2}\|w\|^2$ subject to $y_i\left[\langle x_i, w\rangle + b\right] \geq 1$

$$w = \sum_i y_i \alpha_i x_i$$



Dual $\underset{\alpha}{\text{maximize}} - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j\rangle + \sum_i \alpha_i$

subject to $\sum_i \alpha_i y_i = 0$ and $\alpha_i \geq 0$

Lagrangian

Solving the optimization problem

- Dual problem

$$\underset{\alpha}{\text{maximize}} - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle + \sum_i \alpha_i$$

$$\text{subject to } \sum_i \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0$$

Lagrangian

- If problem is small enough (1000s of variables) we can use off-the-shelf solver (CVXOPT, CPLEX, OOQP, LOQO)
- For larger problem use fact that only SVs matter and solve in blocks (active set method).

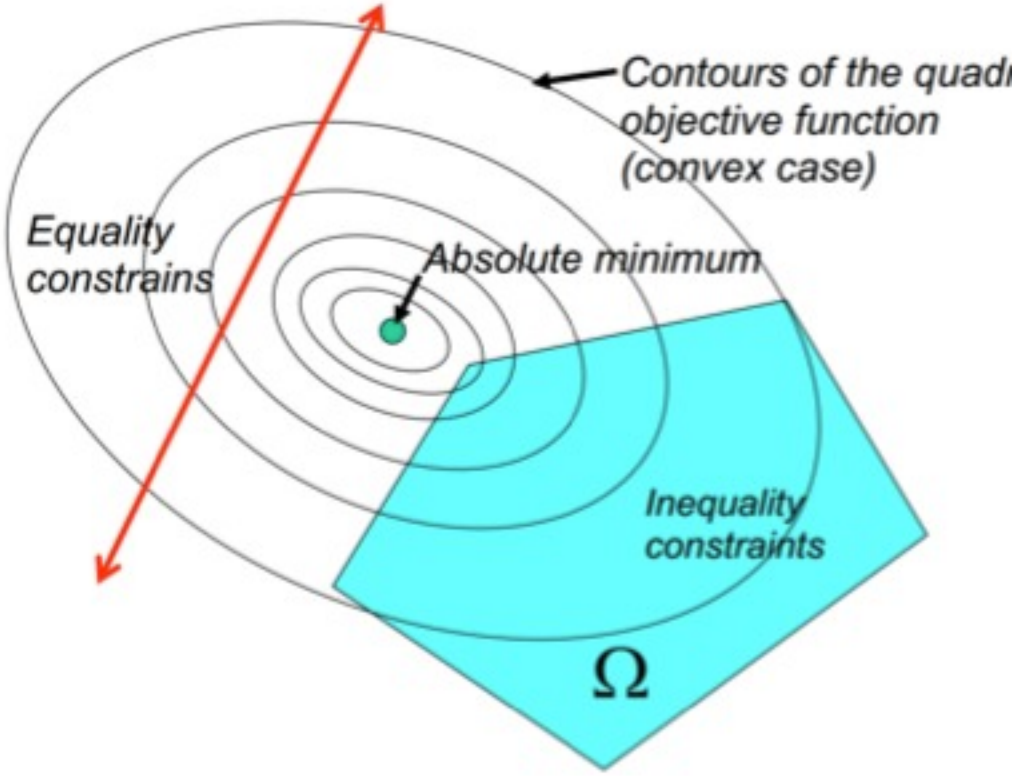
Quadratic Program in Primal

- Primal

$$\min_w \left\{ \frac{1}{2} w^T Q w + c^T w \right\} \text{ subject to } \begin{cases} Aw \leq b \\ Ew = d \end{cases}$$

where $Q \in \mathbb{R}^{n \times n}$ and is symmetric, $w, c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $E \in \mathbb{R}^{p \times n}$, and $d \in \mathbb{R}^p$,

Q: what's the Q in SVM primal?
how about Q in SVM dual?



Quadratic Program in Dual

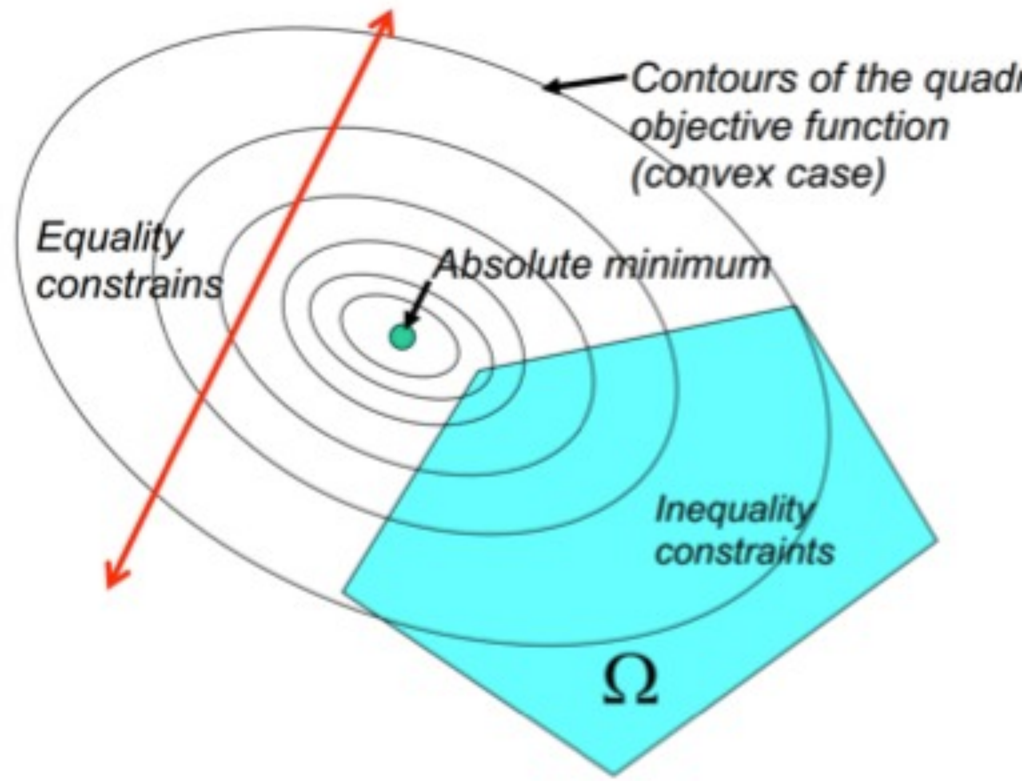
- Dual problem

$$\underset{\alpha}{\text{maximize}} -\frac{1}{2}\alpha^T Q\alpha - \alpha^T b$$
$$\text{subject to } \alpha \geq 0$$

Q: what's the Q in SVM primal?
how about Q in SVM dual?

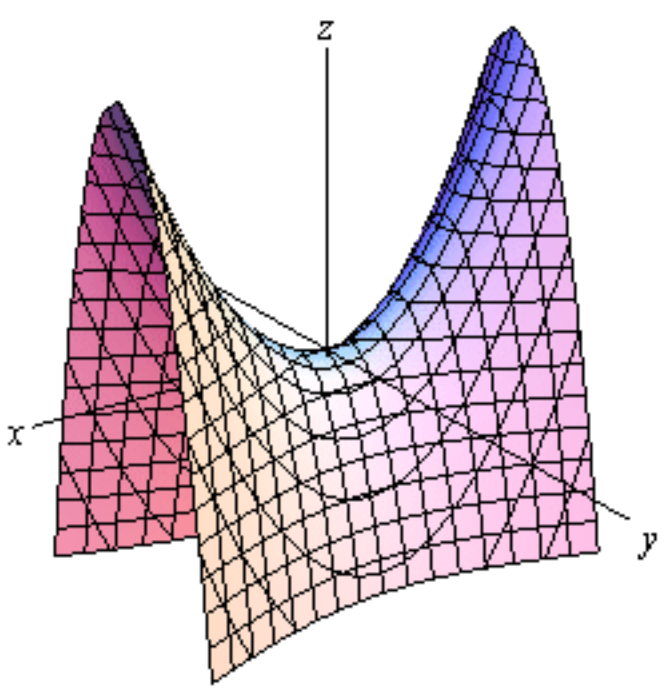
- Quadratic Programming
  - Objective: Quadratic function
    - Q is positive semidefinite
  - Constraints: Linear functions

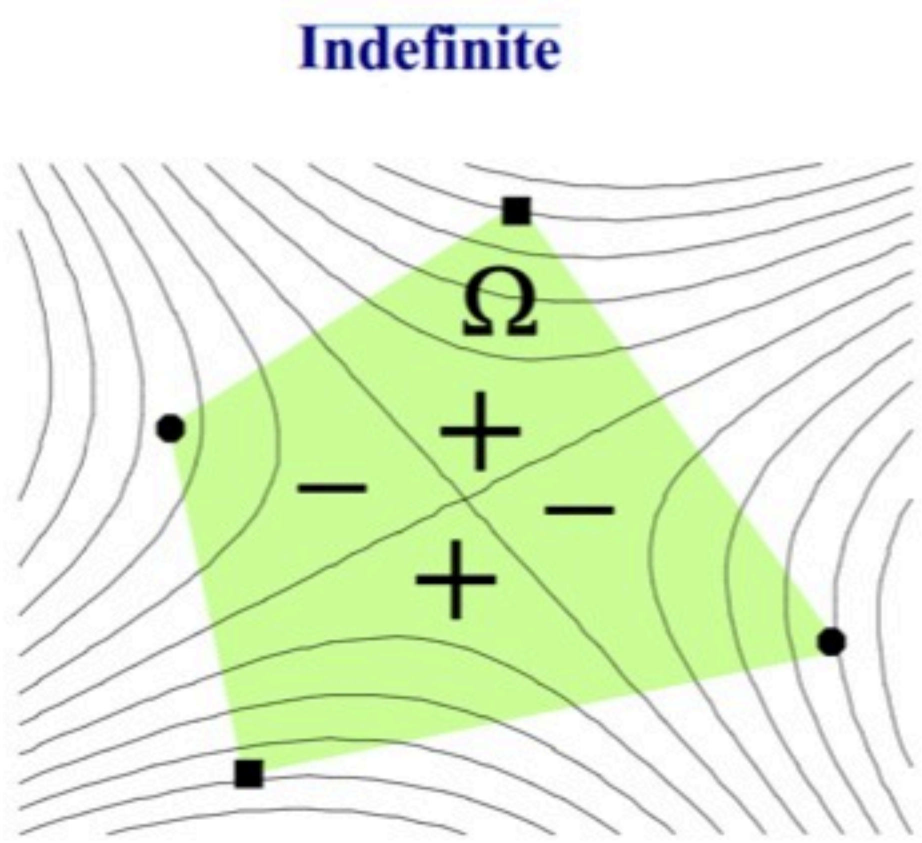
- Methods
  - Gradient Descent
  - Coordinate Descent
    - aka., Hildreth Algorithm
  - Sequential Minimal Optimization (SMO)

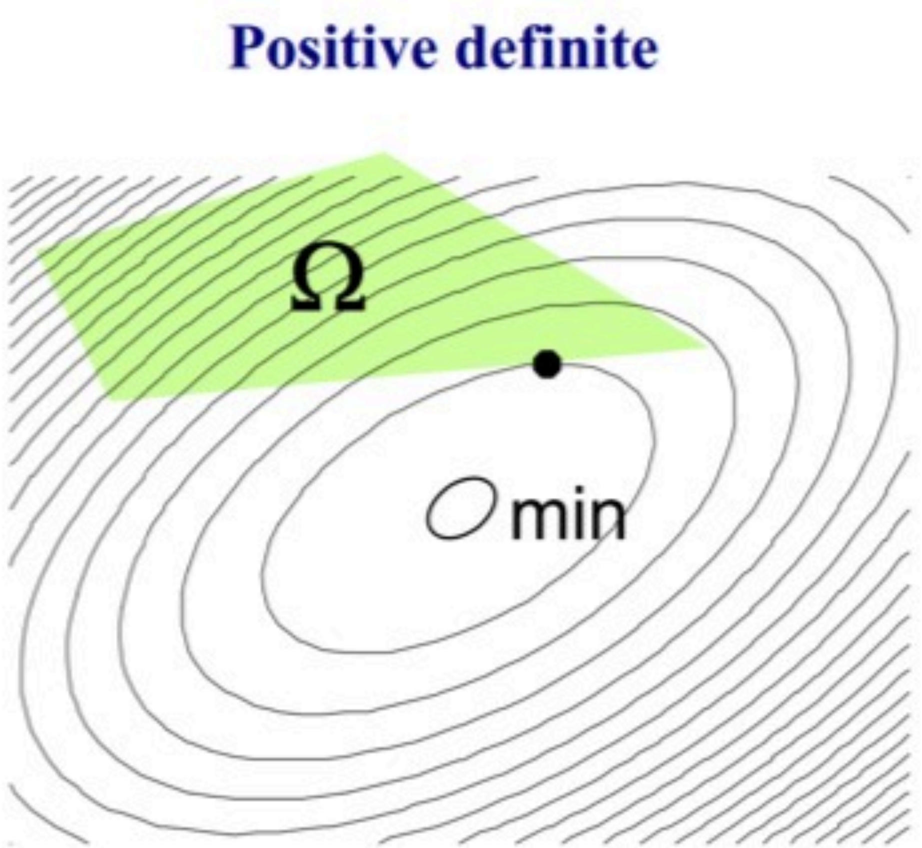


Convex QP

- if Q is positive (semi)definite, i.e., $x^TQx >= 0$, then convex QP => local min/max is global min/max
- if Q = 0, it reduces to linear programming
- general QP is NP-hard; convex QP is polynomial







Hildreth Algorithm

- idea 1:
  - update one coordinate while fixing all other coordinates
    - e.g., update coordinate i is to solve:

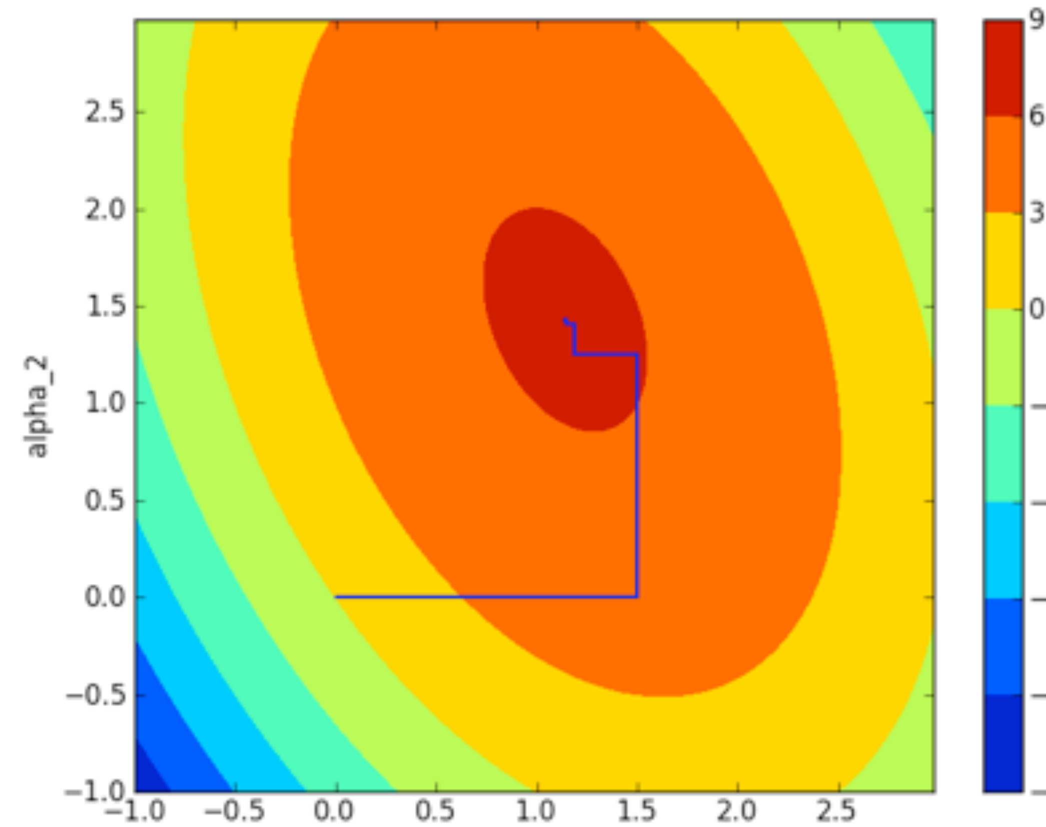
$$\underset{\alpha_i}{\text{argmax}} -\frac{1}{2}\alpha^T Q\alpha - \alpha^T b$$

$$\text{subject to } \alpha \geq 0$$

Quadratic function with only one variable
Maximum => first-order derivative is 0

Hildreth Algorithm

- idea 2:
  - choose another coordinate and repeat until meet stopping criterion
    - reach maximum or
    - increase between 2 consecutive iterations is very small or
    - after some # of iterations
  - how to choose coordinate: sweep patter
    - Sequential:
      - 1, 2, ..., n, 1, 2, ..., n, ...
      - 1, 2, ..., n, n-1, n-2, ...,1, 2, ...
    - Random: permutation of 1,2, ..., n
    - Maximal Descent
      - choose i with maximal descent in objecti



Hildreth Algorithm

initialize $\alpha_i = 0$ for all i

repeat

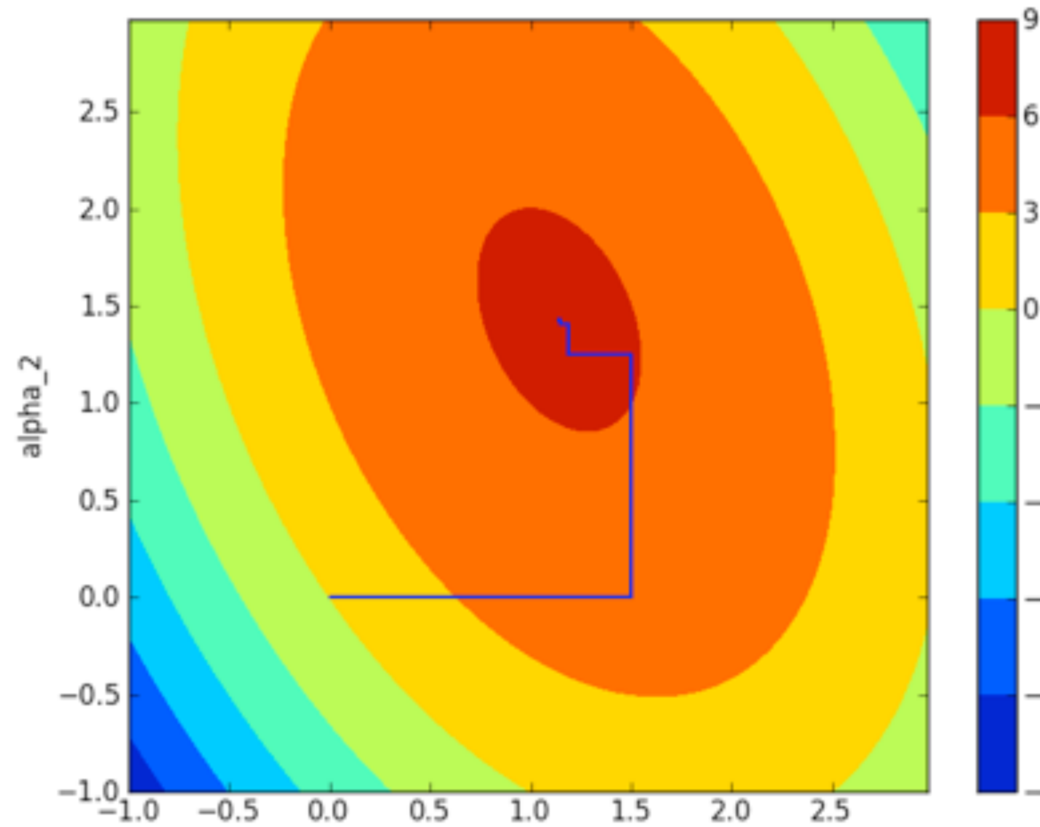
pick i following sweep pattern

solve

$$\alpha_i \leftarrow \operatorname*{argmax}_{\alpha_i} -\frac{1}{2}\alpha^T Q \alpha - \alpha^T b$$

$$\text{subject to } \alpha \geq 0$$

until meet stopping criterion

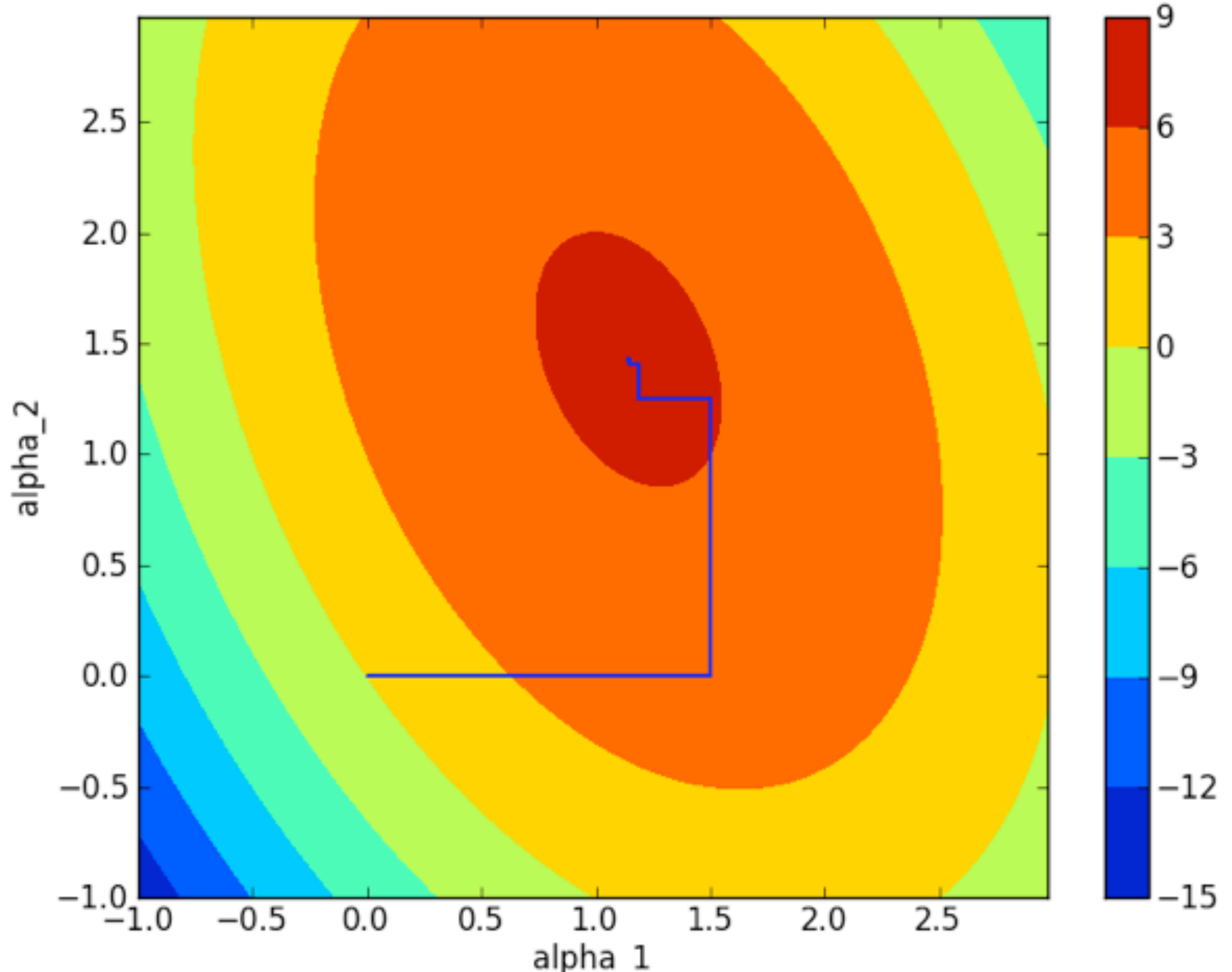


Hildreth Algorithm

$$\underset{\alpha}{\text{maximize}} -\frac{1}{2}\alpha^T \begin{pmatrix} 4 & 1 \\ 1 & 2 \end{pmatrix} \alpha - \alpha^T \begin{pmatrix} -6 \\ -4 \end{pmatrix}$$

$$\text{subject to } \alpha \geq 0$$

- choose coordinates
  - 1, 2, 1, 2, ...

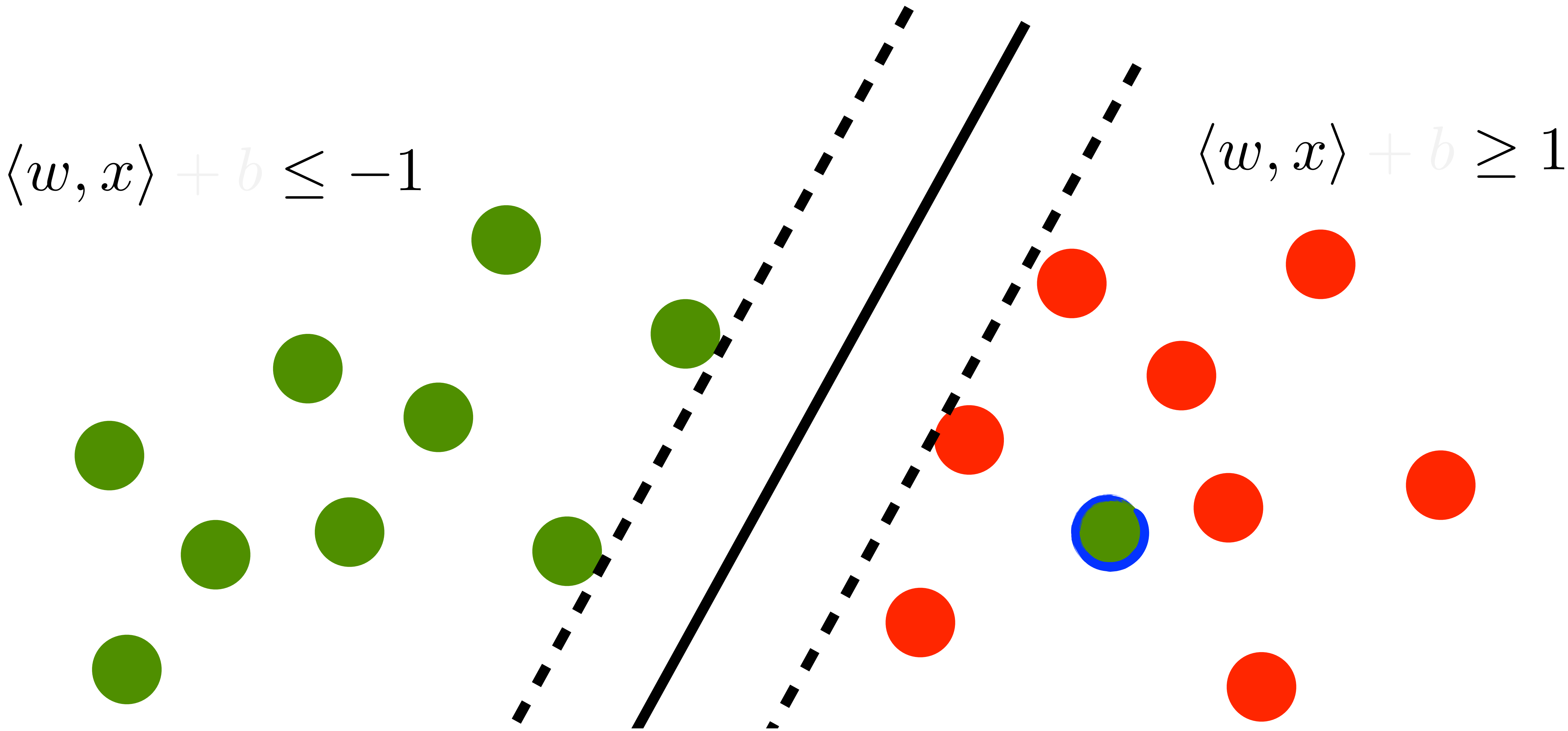


Hildreth Algorithm

- pros:
  - extremely simple
  - no gradient calculation
  - easy to implement
- cons:
  - converges slow, compared to other methods

Carnegie Mellon University

Large Margin Classifier

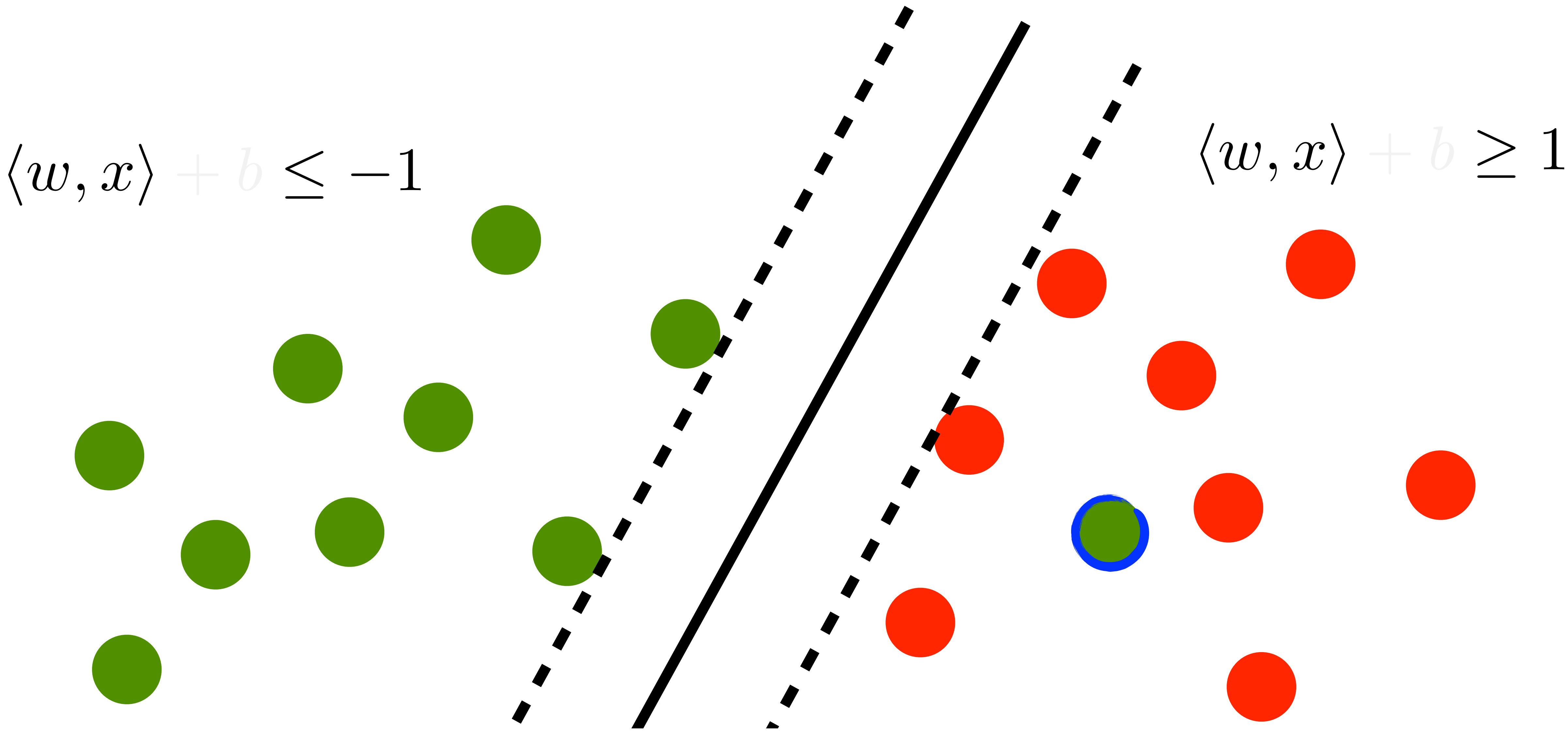


linear function

$f(x) = \langle w, x \rangle + b$

linear separator
is impossible

Large Margin Classifier



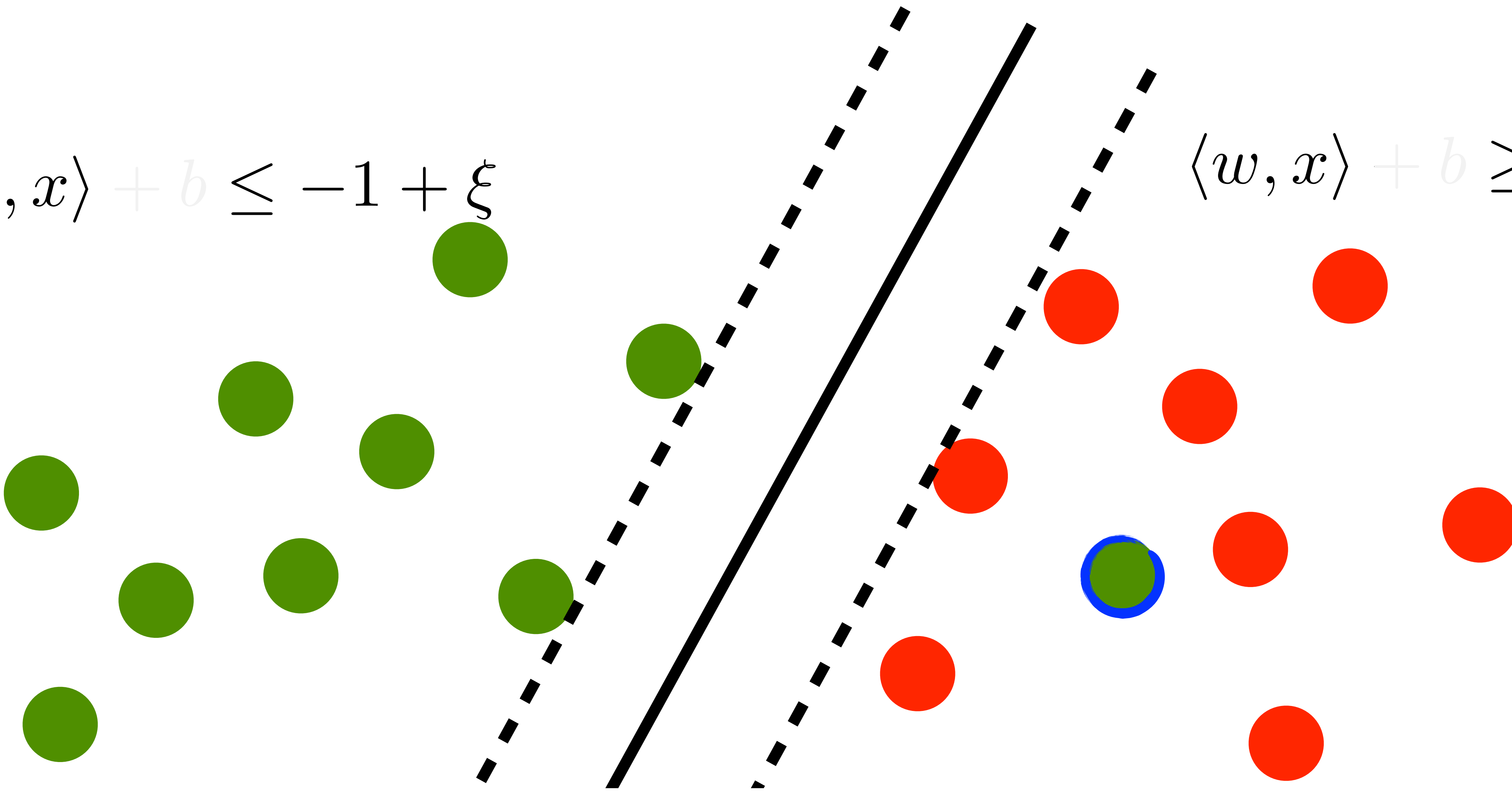
minimum error separator
is impossible

Theorem (Minsky & Papert)
Finding the minimum error separating hyperplane is NP hard

Adding slack variables

$\langle w, x \rangle + b \leq -1 + \xi$

$\langle w, x \rangle + b \geq 1 - \xi$



minimize amount of slack

Convex optimization problem

Adding slack variables

- Hard margin problem

$$\underset{w,b}{\text{minimize}} \frac{1}{2}\|w\|^2 \text{ subject to } y_i \left[\langle w, x_i \rangle + b\right] \geq 1$$

- With slack variables

$$\underset{w,b}{\text{minimize}} \ \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$\text{subject to } y_i \left[\langle w, x_i \rangle + b\right] \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

Problem is always feasible. Proof:

$w = 0$ and $b = 0$ and $\xi_i = 1$ (also yields upper bound)

Intermezzo
Convex Programs for Dummies

- Primal optimization problem

$$\underset{x}{\text{minimize}}\, f(x) \text{ subject to } c_i(x) \leq 0$$

- Lagrange function

$$L(x, \alpha) = f(x) + \sum_i \alpha_i c_i(x)$$

- First order optimality conditions in x

$$\partial_x L(x, \alpha) = \partial_x f(x) + \sum_i \alpha_i \partial_x c_i(x) = 0$$

- Solve for x and plug it back into L

$$\underset{\alpha}{\text{maximize}}\, L(x(\alpha), \alpha)$$

(keep explicit constraints)

Dual Problem

- Primal optimization problem

$$\underset{w,b}{\text{minimize}} \ \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$\text{subject to } y_i \left[\langle w, x_i\rangle + b\right] \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

- Lagrange function

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i - \sum_i \alpha_i \left[y_i \left[\langle x_i, w\rangle + b\right] + \xi_i - 1\right] - \sum_i \eta_i \xi_i$$

Optimality in w, ξ is at saddle point with α, η

- Derivatives in w, ξ need to vanish

Dual Problem

- **Lagrange function**

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i - \sum_i \alpha_i \left[y_i \left[\langle x_i, w\rangle + b\right] + \xi_i - 1\right] - \sum_i \eta_i \xi_i$$

- **Derivatives in w need to vanish**

$$\partial_w L(w, b, \xi, \alpha, \eta) = w - \sum_i \alpha_i y_i x_i = 0$$

$$\partial_b L(w, b, \xi, \alpha, \eta) = \sum_i \alpha_i y_i = 0$$

$$\partial_{\xi_i} L(w, b, \xi, \alpha, \eta) = C - \alpha_i - \eta_i = 0$$

- **Plugging terms back into L yields**

$$\underset{\alpha}{\text{maximize}} - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j\rangle + \sum_i \alpha_i$$

subject to $\sum_i \alpha_i y_i = 0$ and $\alpha_i \in [0, C]$

Lagrangian

bound influence

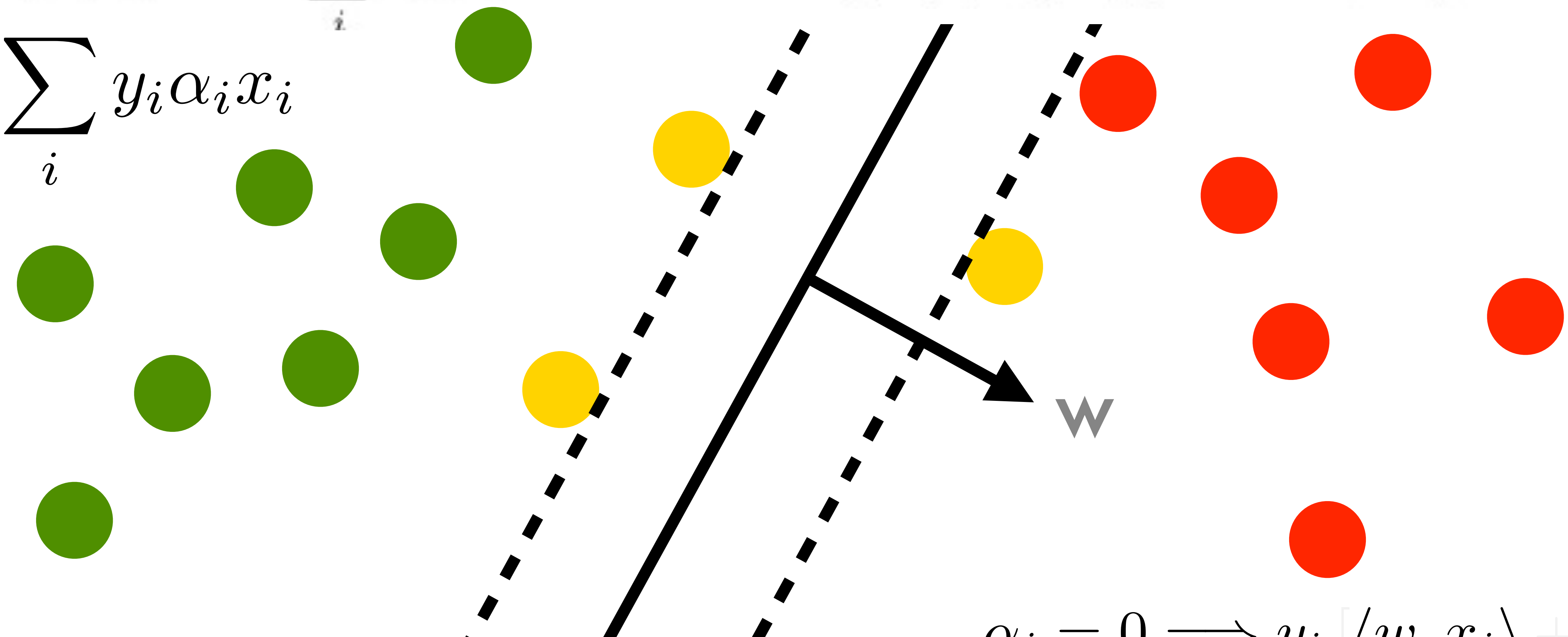
Karush Kuhn Tucker Conditions

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i - \sum_i \alpha_i \left[y_i \left[\langle x_i, w\rangle + b\right] + \xi_i - 1\right] - \sum_i \eta_i \xi_i$$

$$\partial_w L(w, b, \xi, \alpha, \eta) = w - \sum_i \alpha_i y_i x_i = 0$$

$$\partial_{\xi_i} L(w, b, \xi, \alpha, \eta) = C - \alpha_i - \eta_i = 0$$

$$w = \sum_i y_i \alpha_i x_i$$



$$\alpha_i \left[y_i \left[\langle w, x_i\rangle + b\right] + \xi_i - 1\right] = 0$$

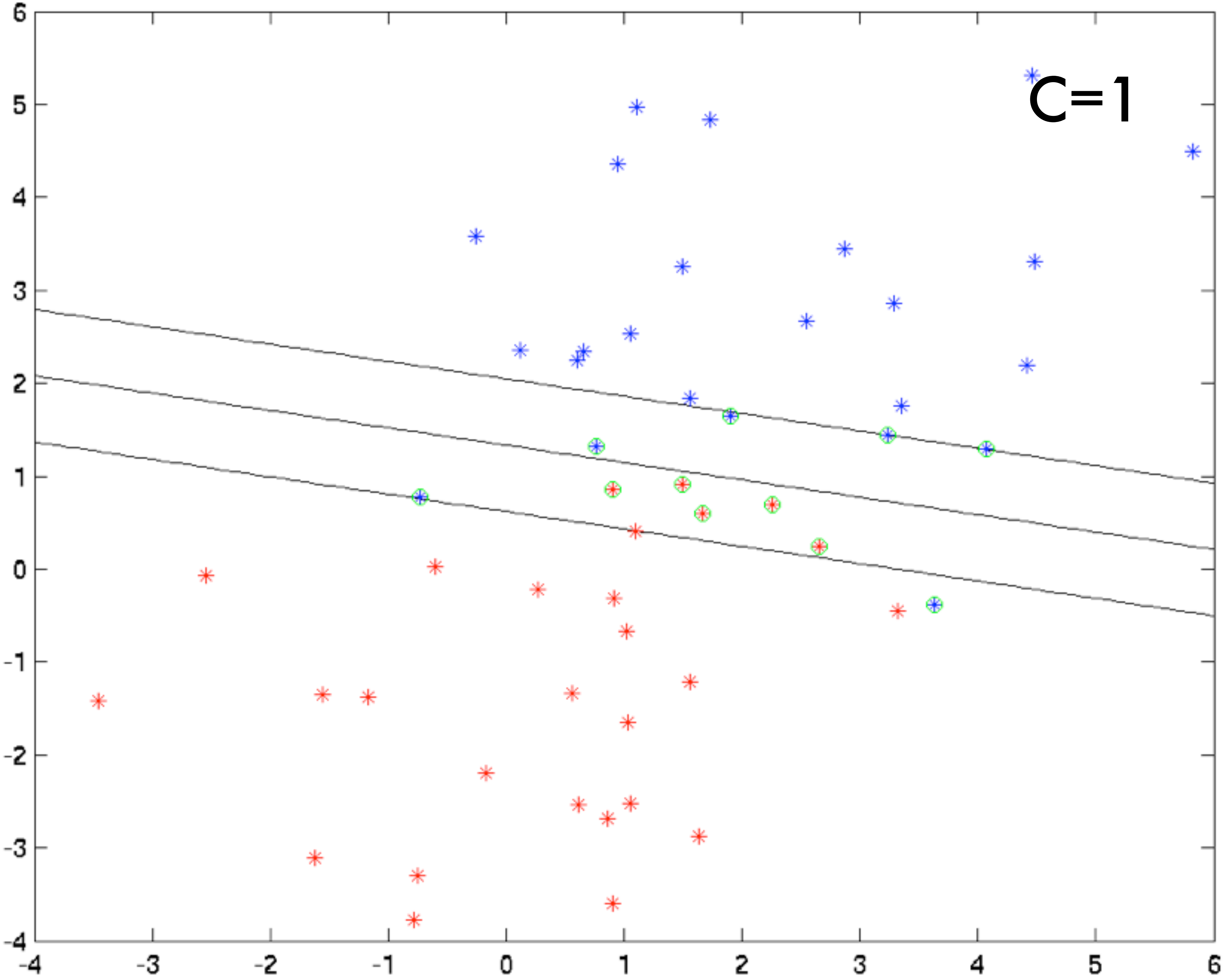
$$\eta_i \xi_i = 0$$

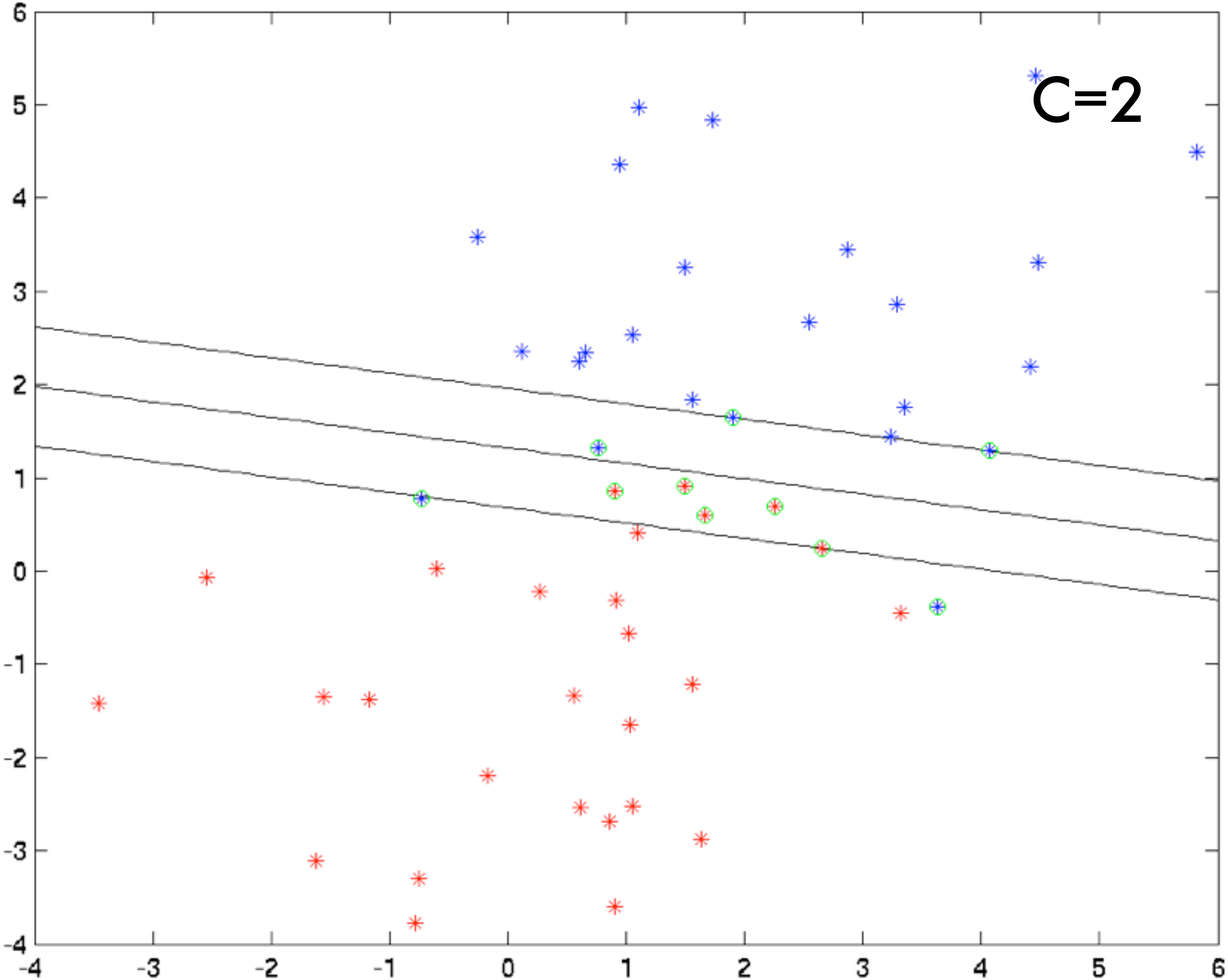
$$0 \leq \alpha_i = C - \eta_i \leq C$$

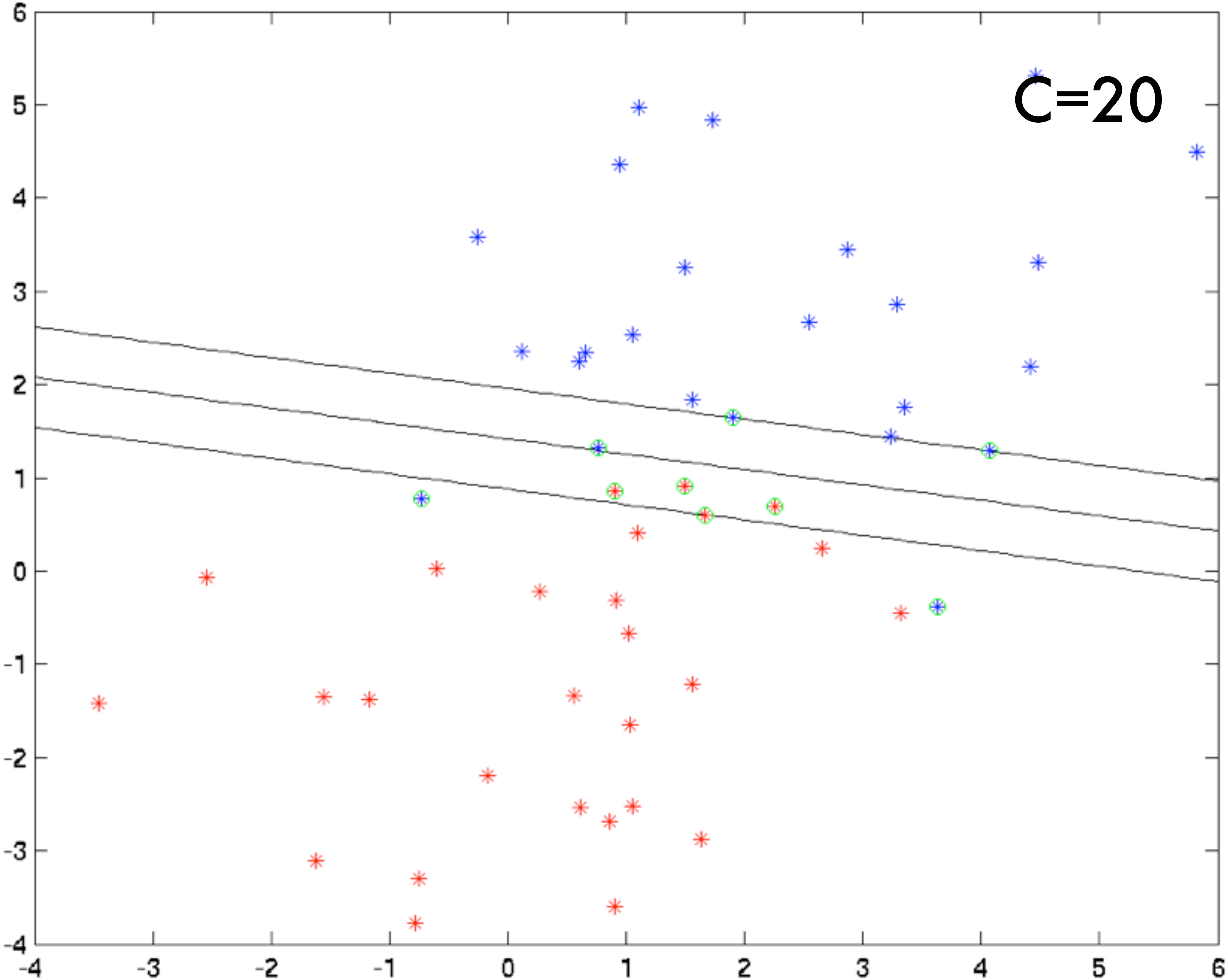
$$\alpha_i = 0 \Longrightarrow y_i \left[\langle w, x_i\rangle + b\right] \geq 1$$

$$0 < \alpha_i < C \Longrightarrow y_i \left[\langle w, x_i\rangle + b\right] = 1$$

$$\alpha_i = C \Longrightarrow y_i \left[\langle w, x_i\rangle + b\right] \leq 1$$


C=1
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6


C=2
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6


C=20
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

Solving the optimization problem

- Dual problem

$$\underset{\alpha}{\text{maximize}} - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i$$

$$\text{subject to } \sum_i \alpha_i y_i = 0 \text{ and } \alpha_i \in [0, C]$$

Lagrangian

- If problem is small enough (1000s of variables) we can use off-the-shelf solver (CVXOPT, CPLEX, OOQP, LOQO) or Hildreth
- For larger problem use fact that only SVs matter and solve in blocks (active set method).


MAGIC Etch A Sketch® SCREEN
Nonlinear
Separation

The Kernel Trick

- **Linear soft margin problem**

$$\underset{w,b}{\text{minimize}} \ \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$\text{subject to } y_i\left[\langle w, x_i\rangle + b\right] \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

- **Dual problem**

$$\underset{\alpha}{\text{maximize}} - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle + \sum_i \alpha_i$$

$$\text{subject to } \sum_i \alpha_i y_i = 0 \text{ and } \alpha_i \in [0, C]$$

Lagrangian

- **Support vector expansion**

$$f(x) = \sum_i \alpha_i y_i \langle x_i, x\rangle + b$$

The Kernel Trick

- **Linear soft margin problem**

$$\underset{w,b}{\text{minimize}} \;\; \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$\text{subject to } y_i\left[\langle w, \phi(x_i)\rangle + b\right] \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

- **Dual problem**

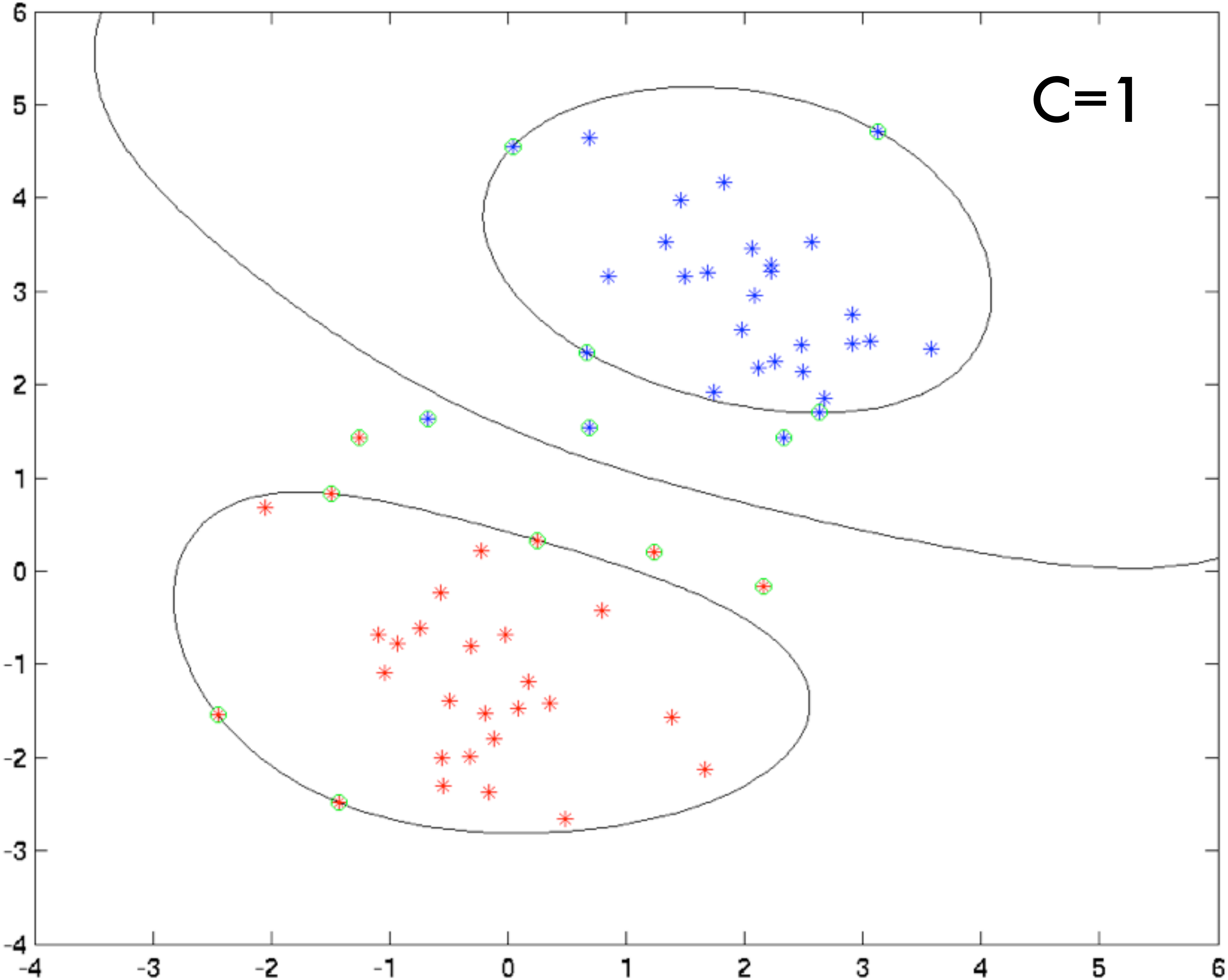
$$\underset{\alpha}{\text{maximize}} - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j k(x_i, x_j) + \sum_i \alpha_i$$

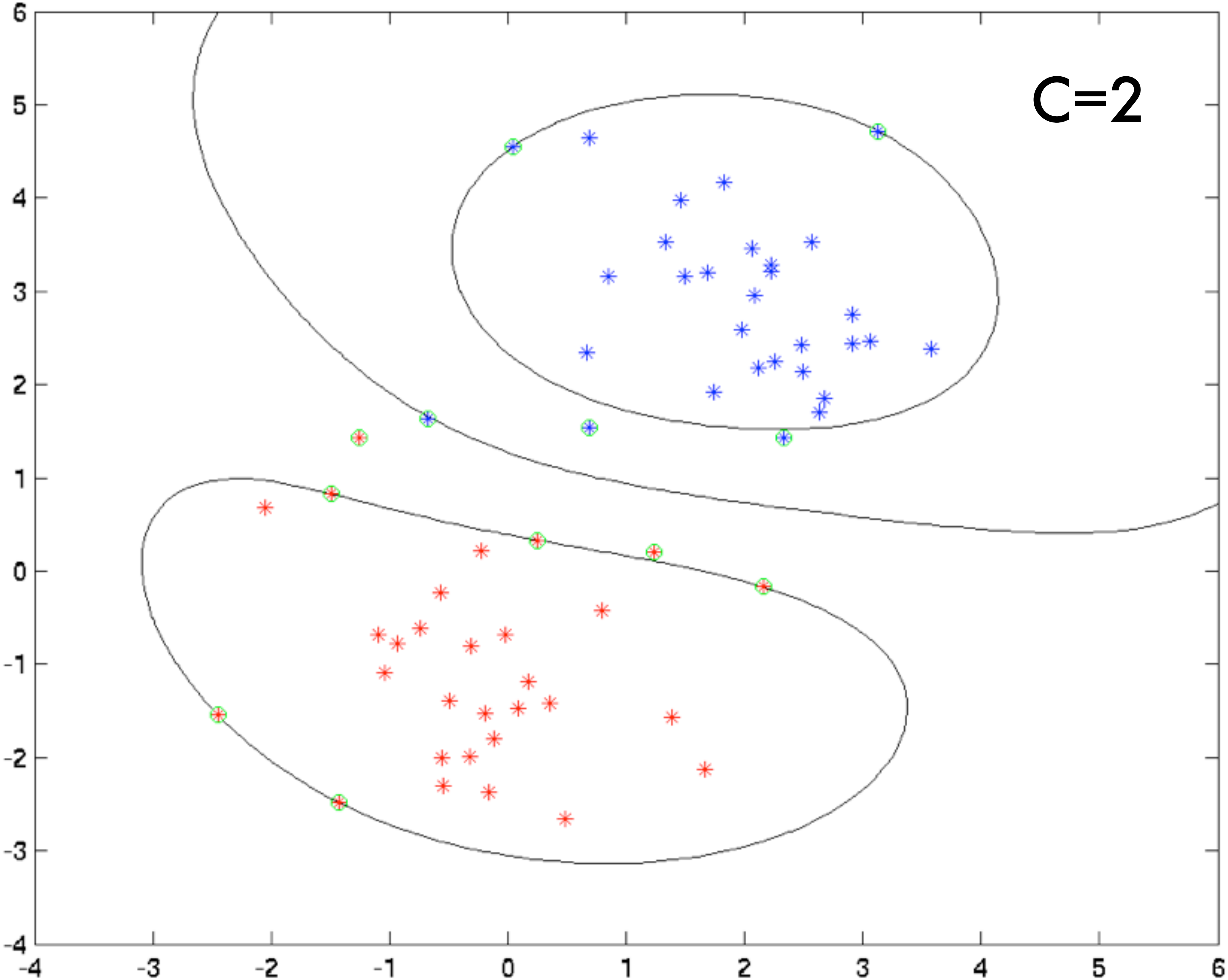
$$\text{subject to } \sum_i \alpha_i y_i = 0 \text{ and } \alpha_i \in [0, C]$$

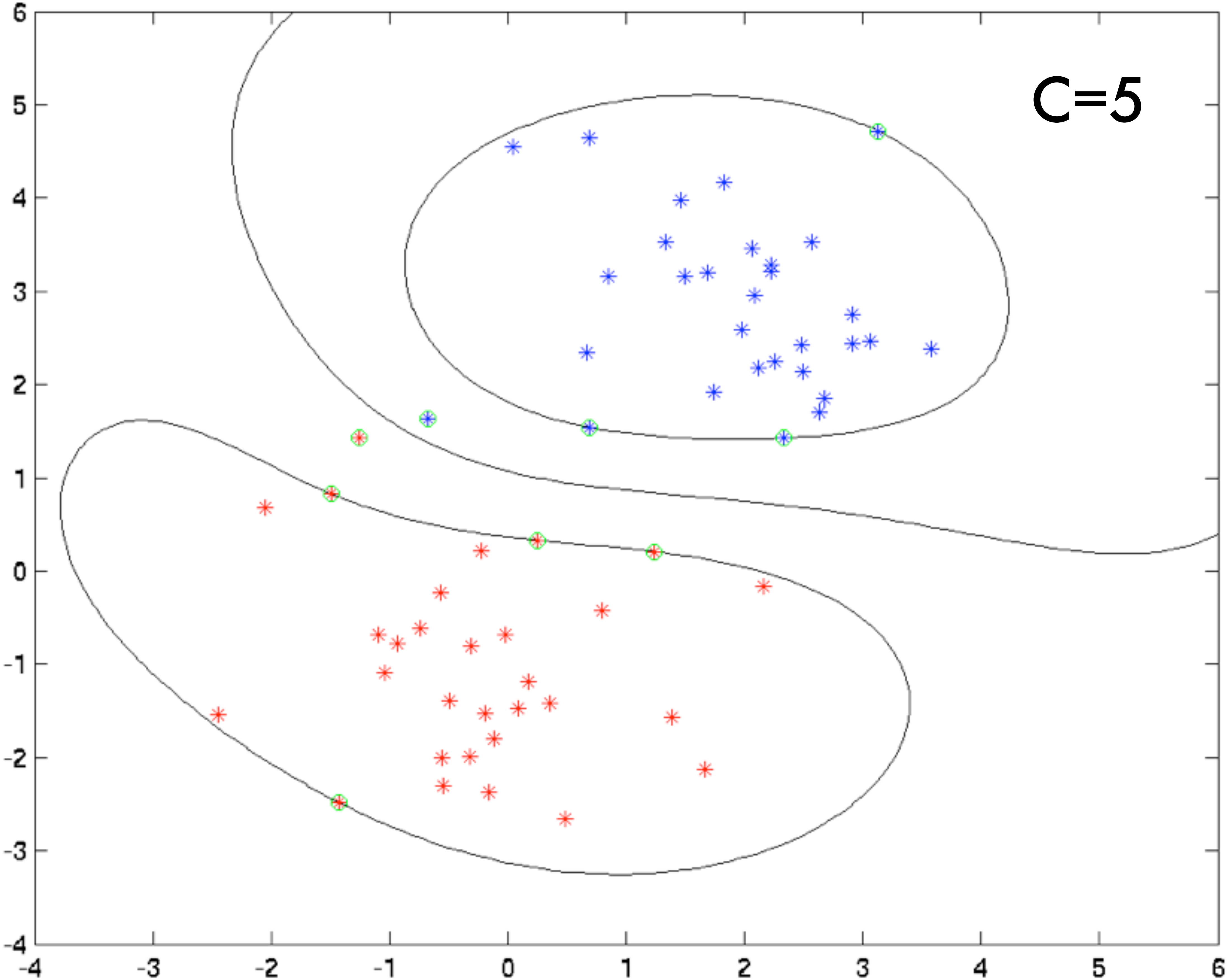
Lagrangian

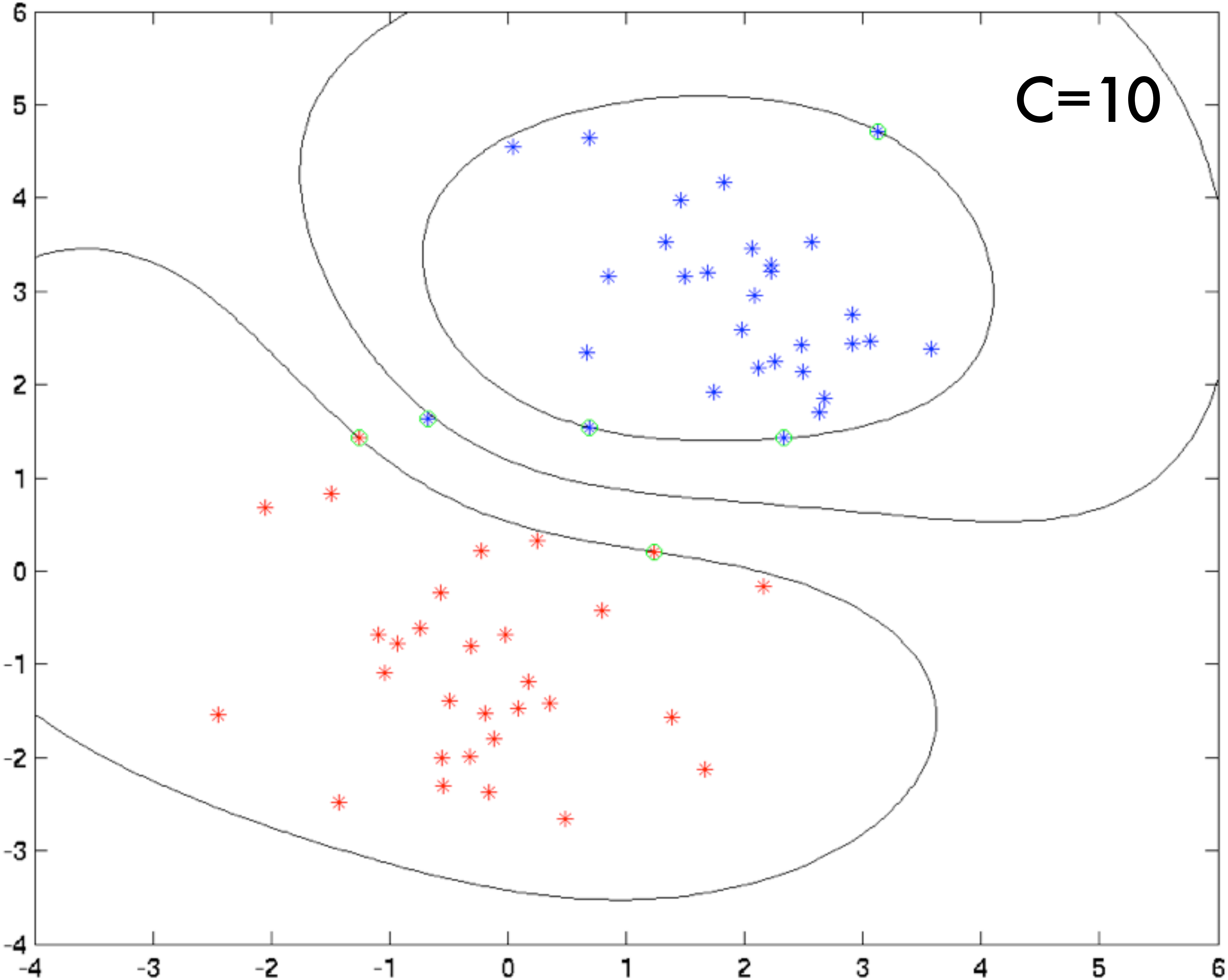
- **Support vector expansion**

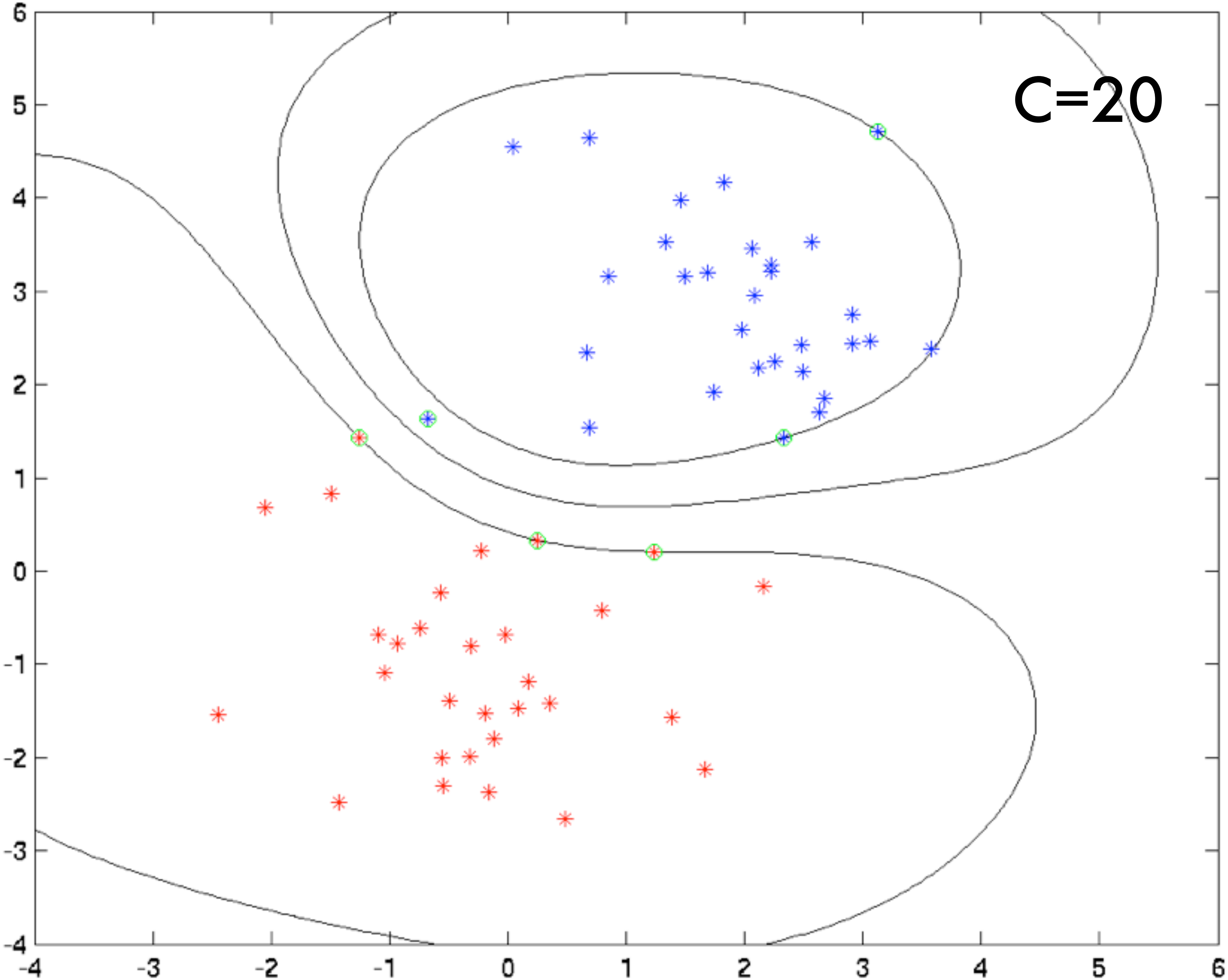
$$f(x) = \sum_i \alpha_i y_i k(x_i, x) + b$$

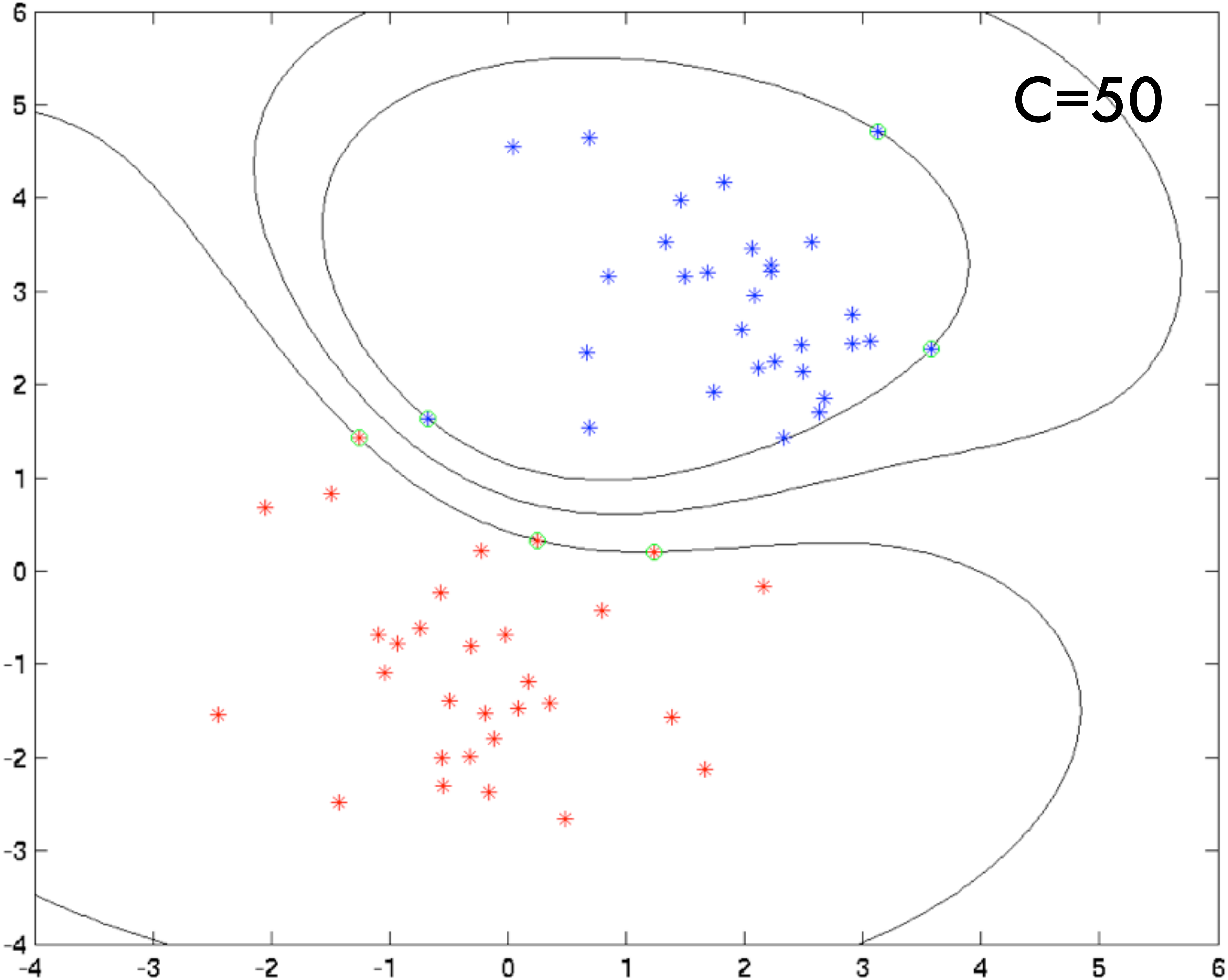

C=1

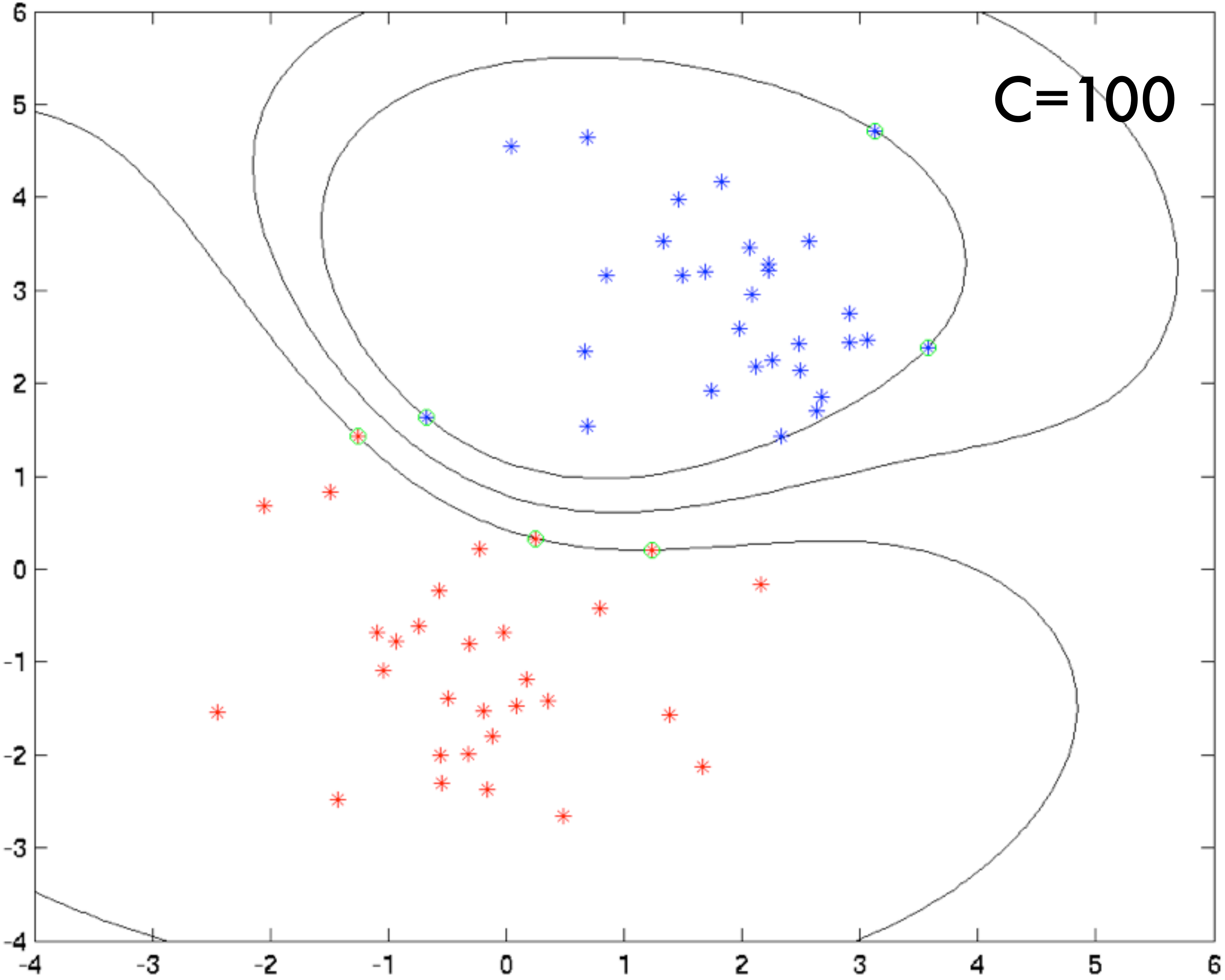

C=2
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

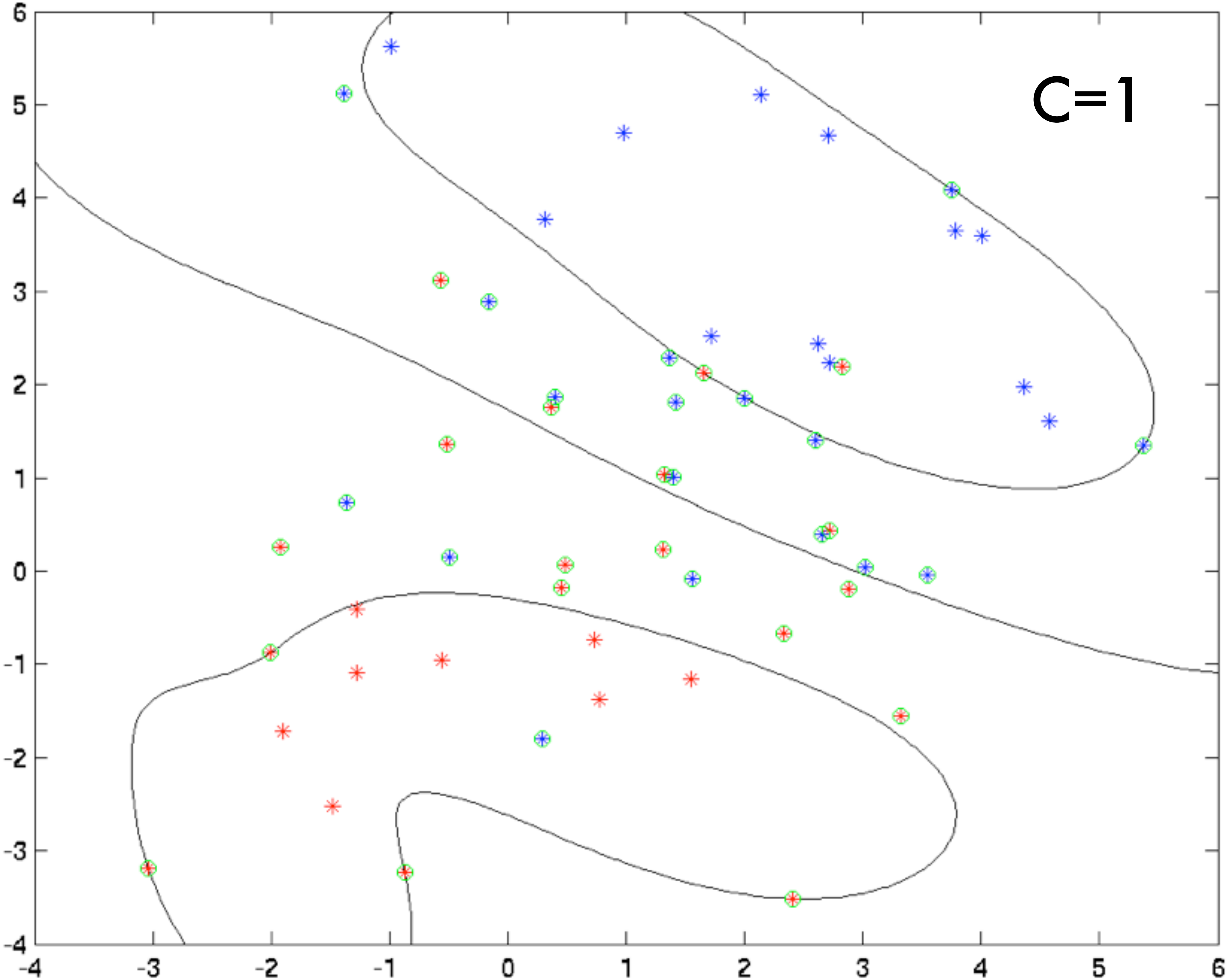

C=5
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

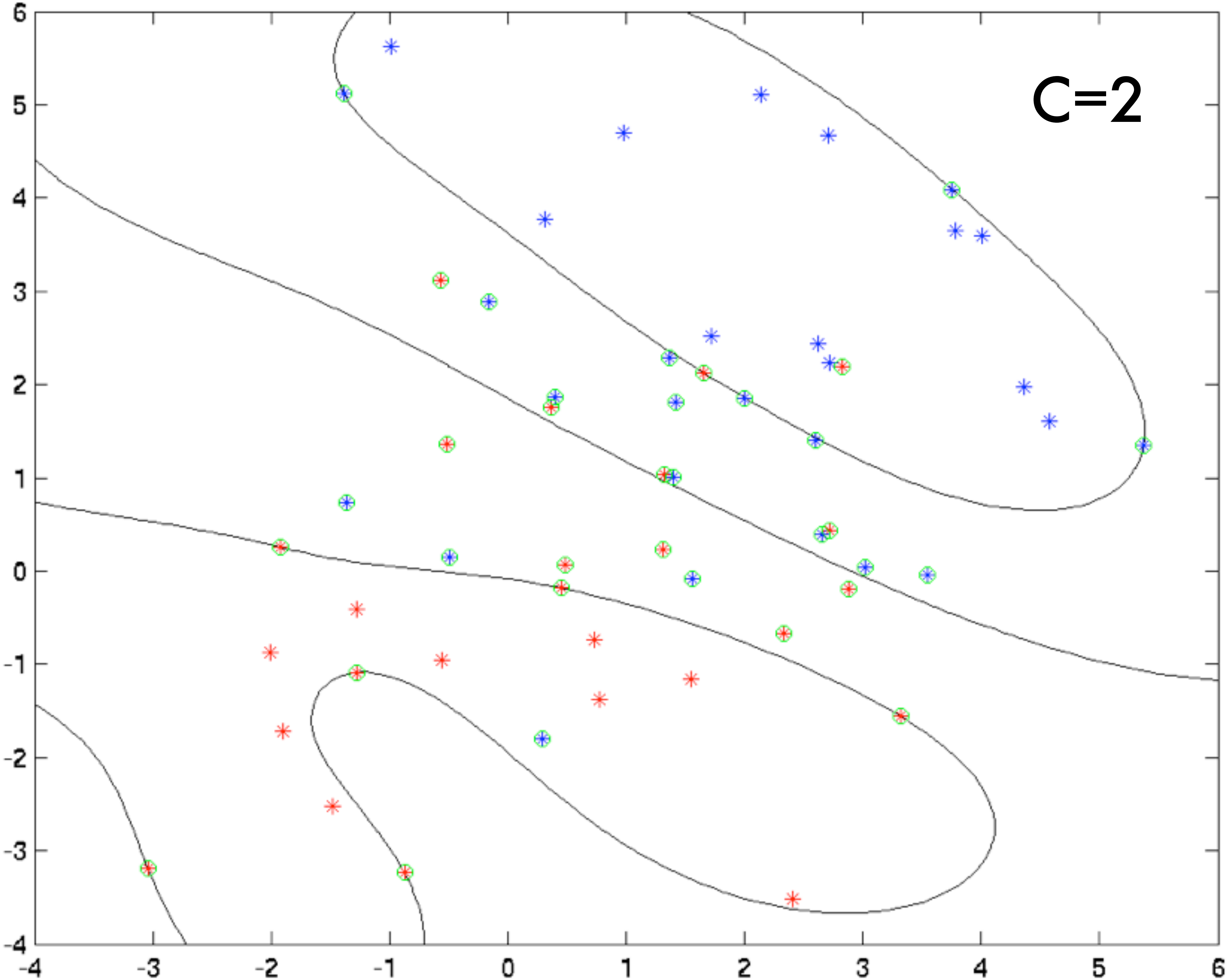

C=10
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

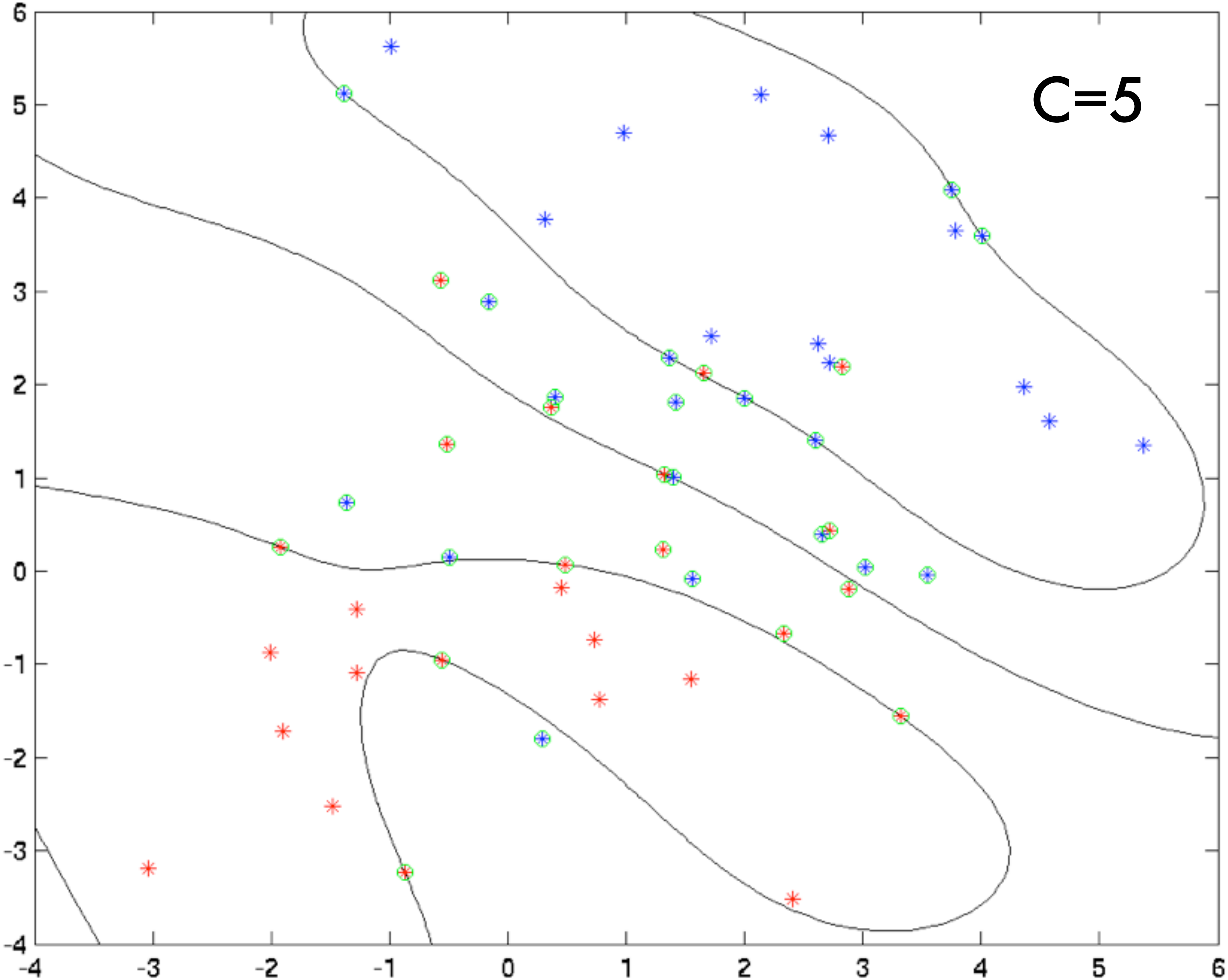

C=20
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

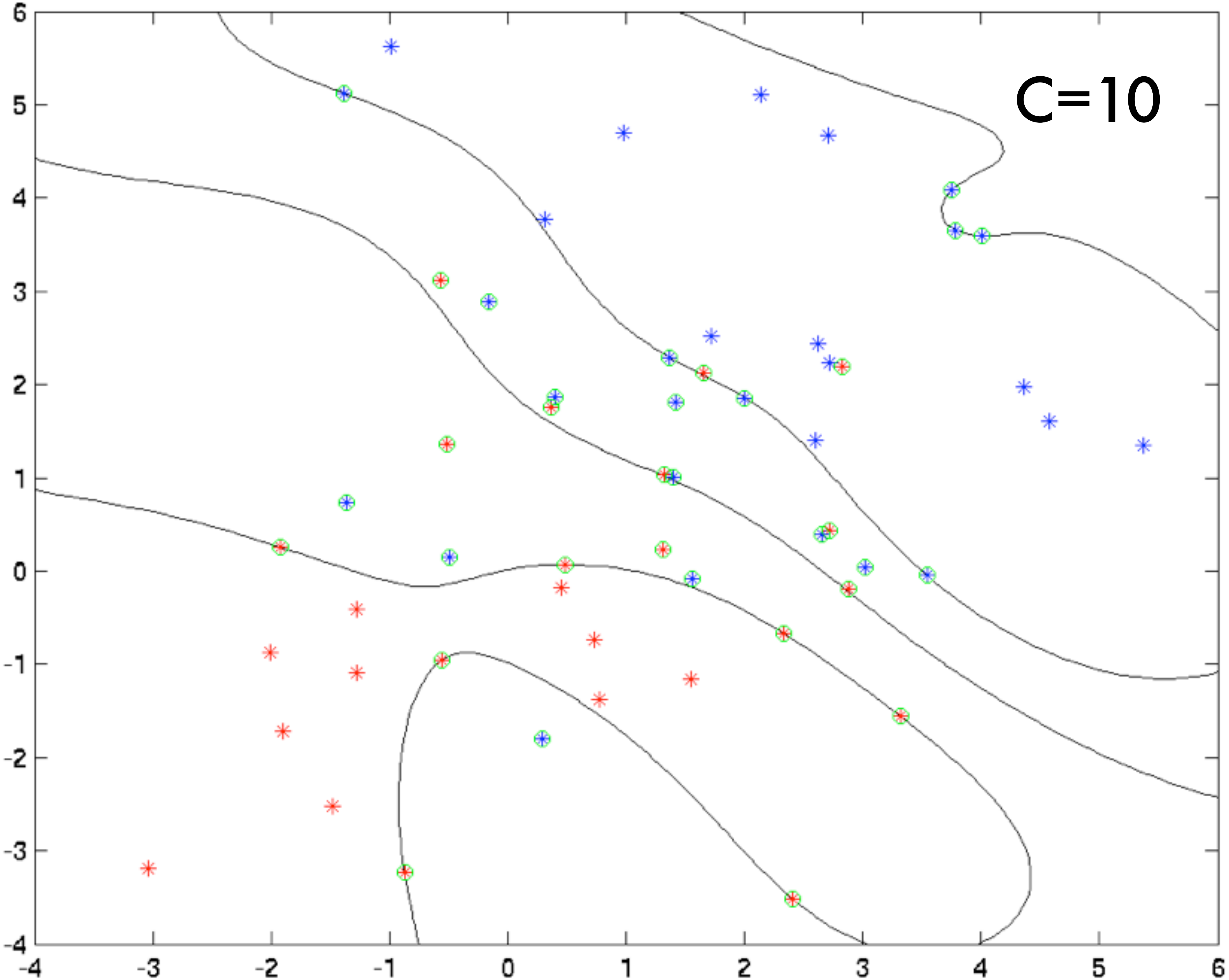

C=50

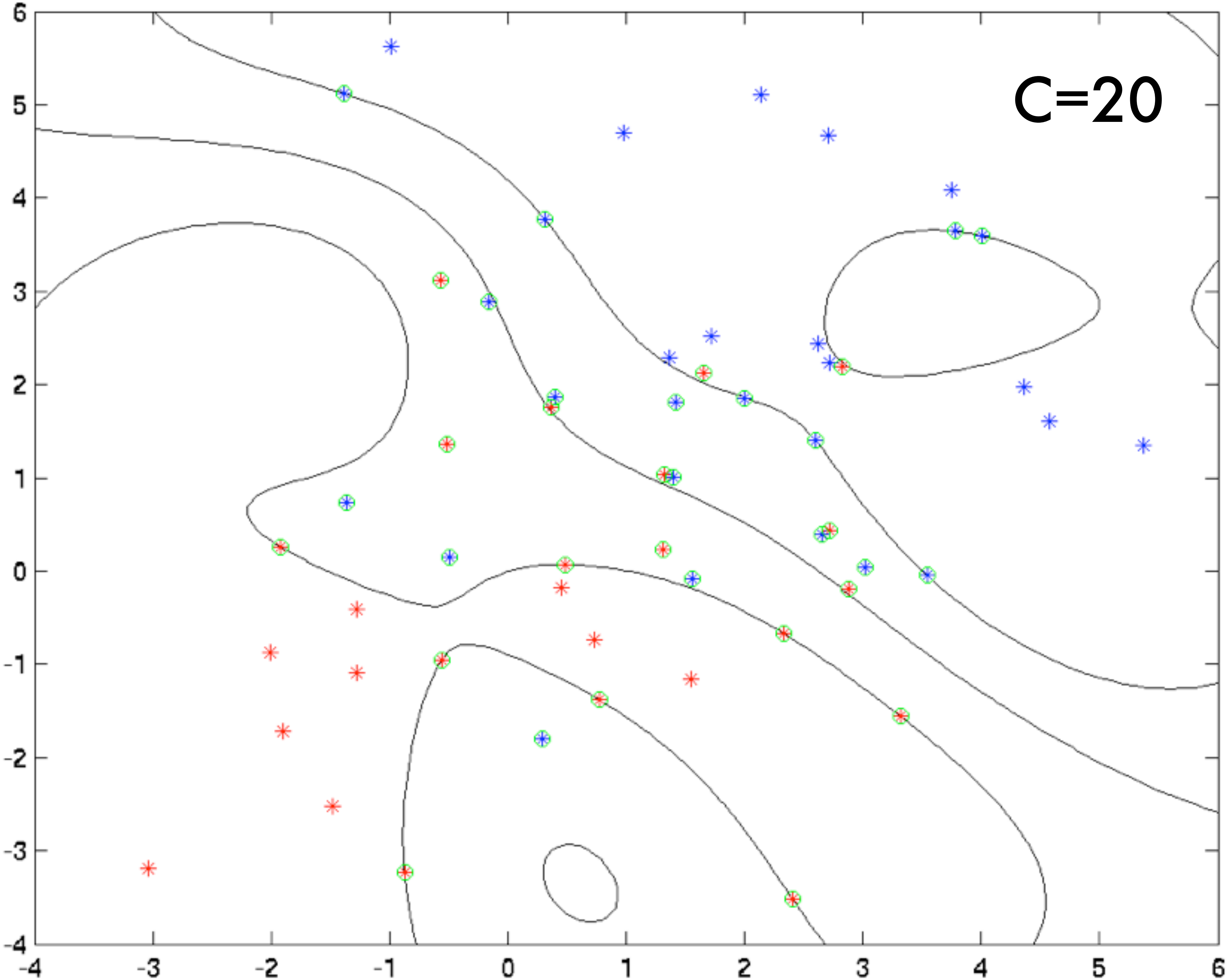

C=100
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

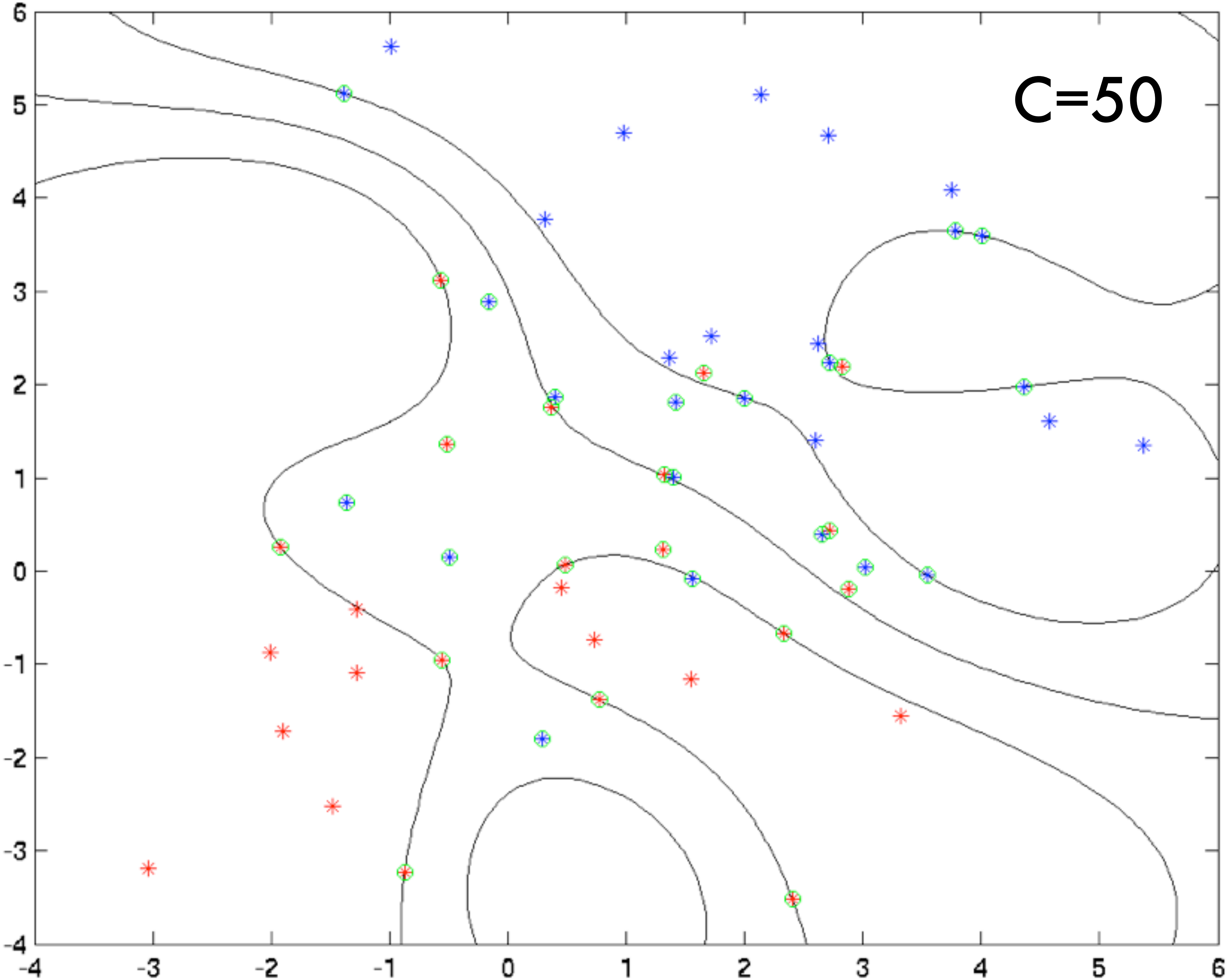

C=1
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

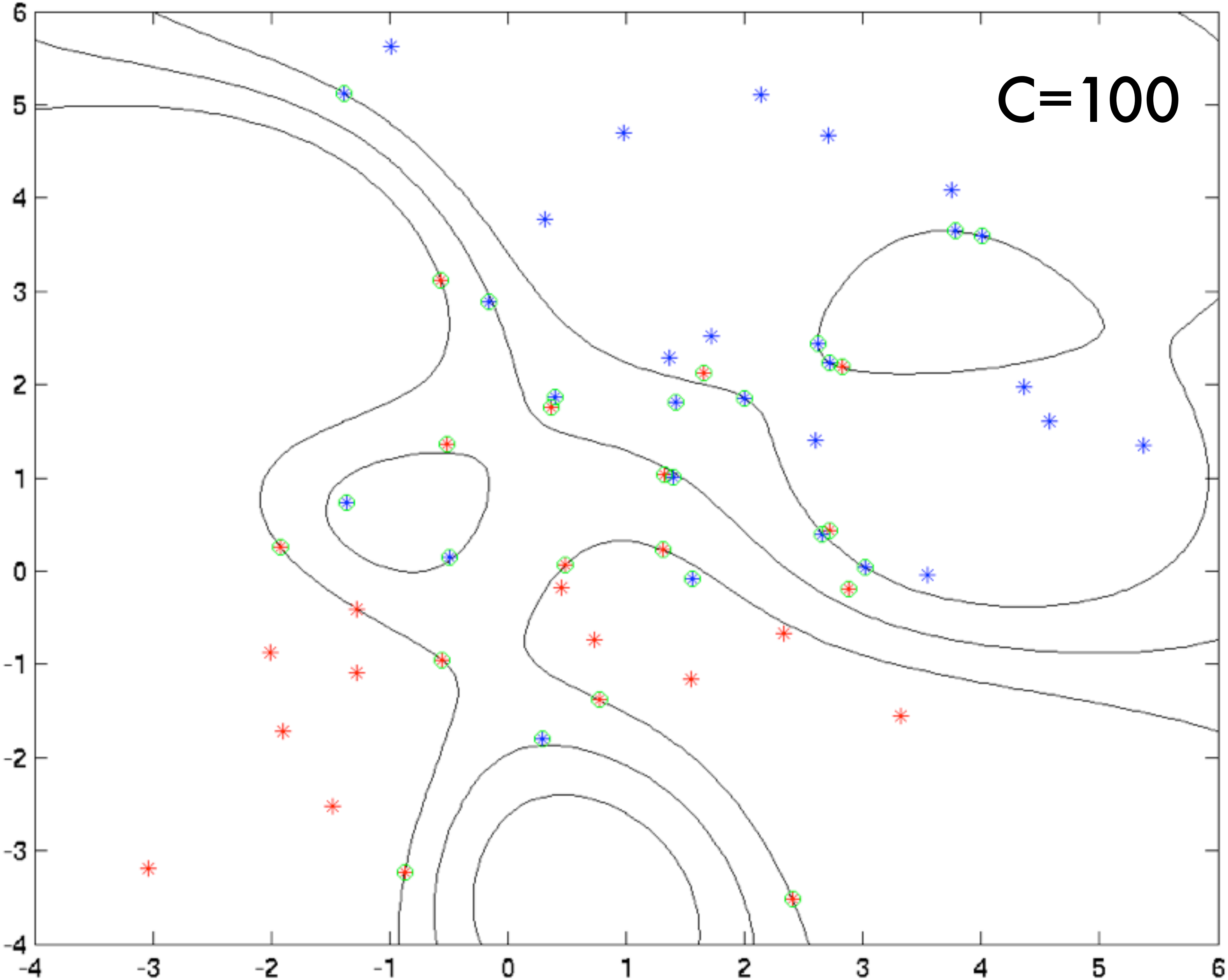

C=2
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6


C=5
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6


C=10
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

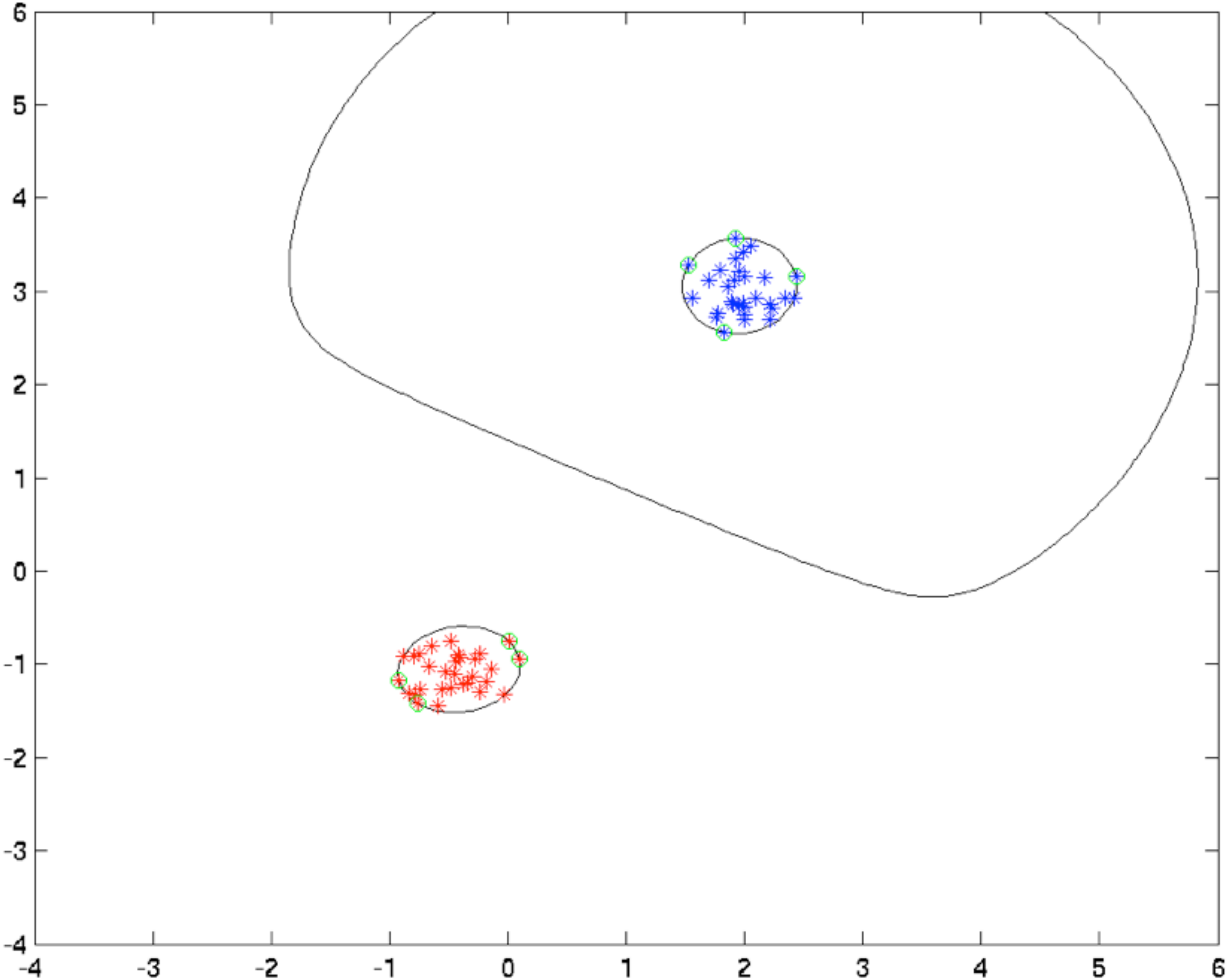

C=20
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6


C=50
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

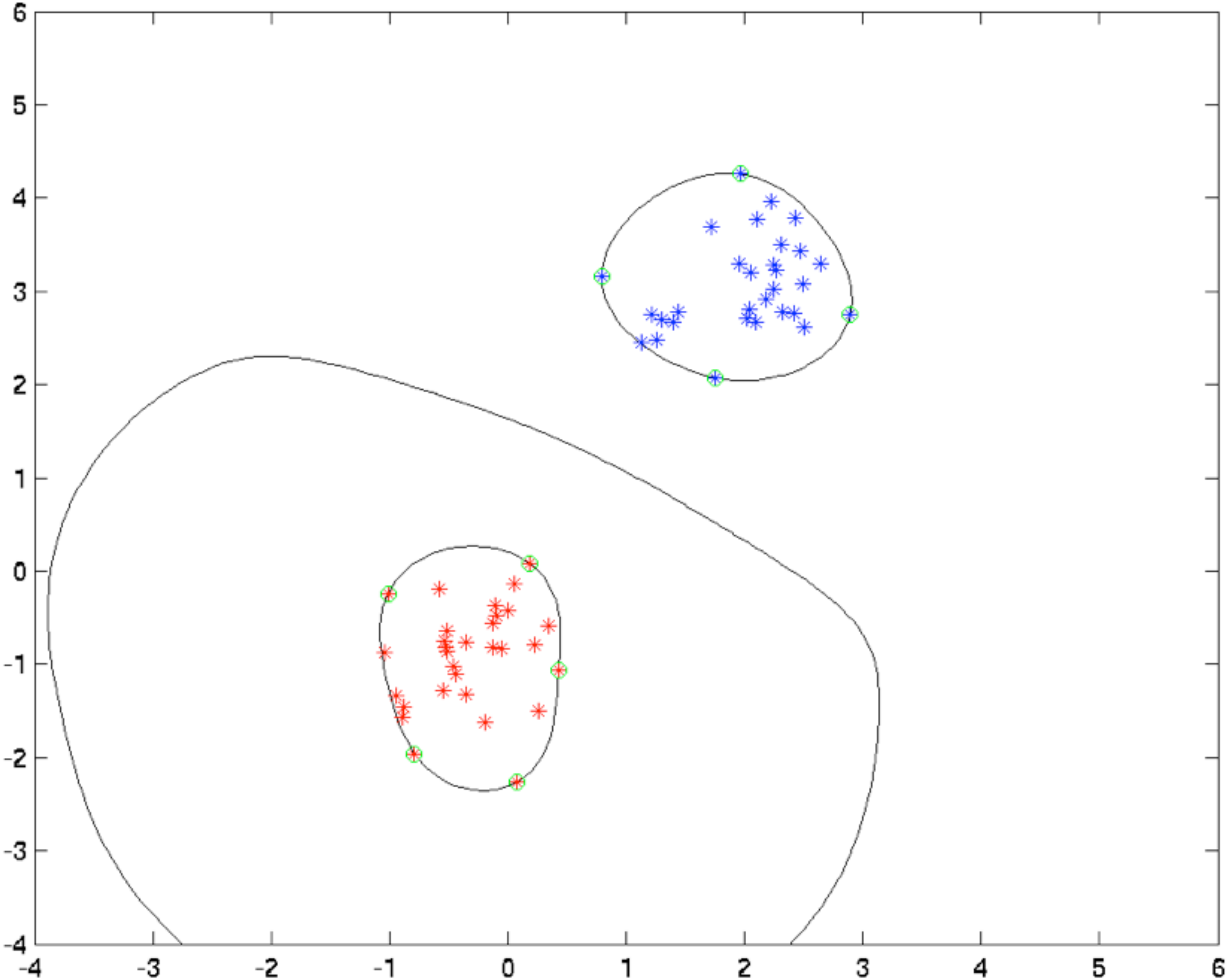


C=100
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

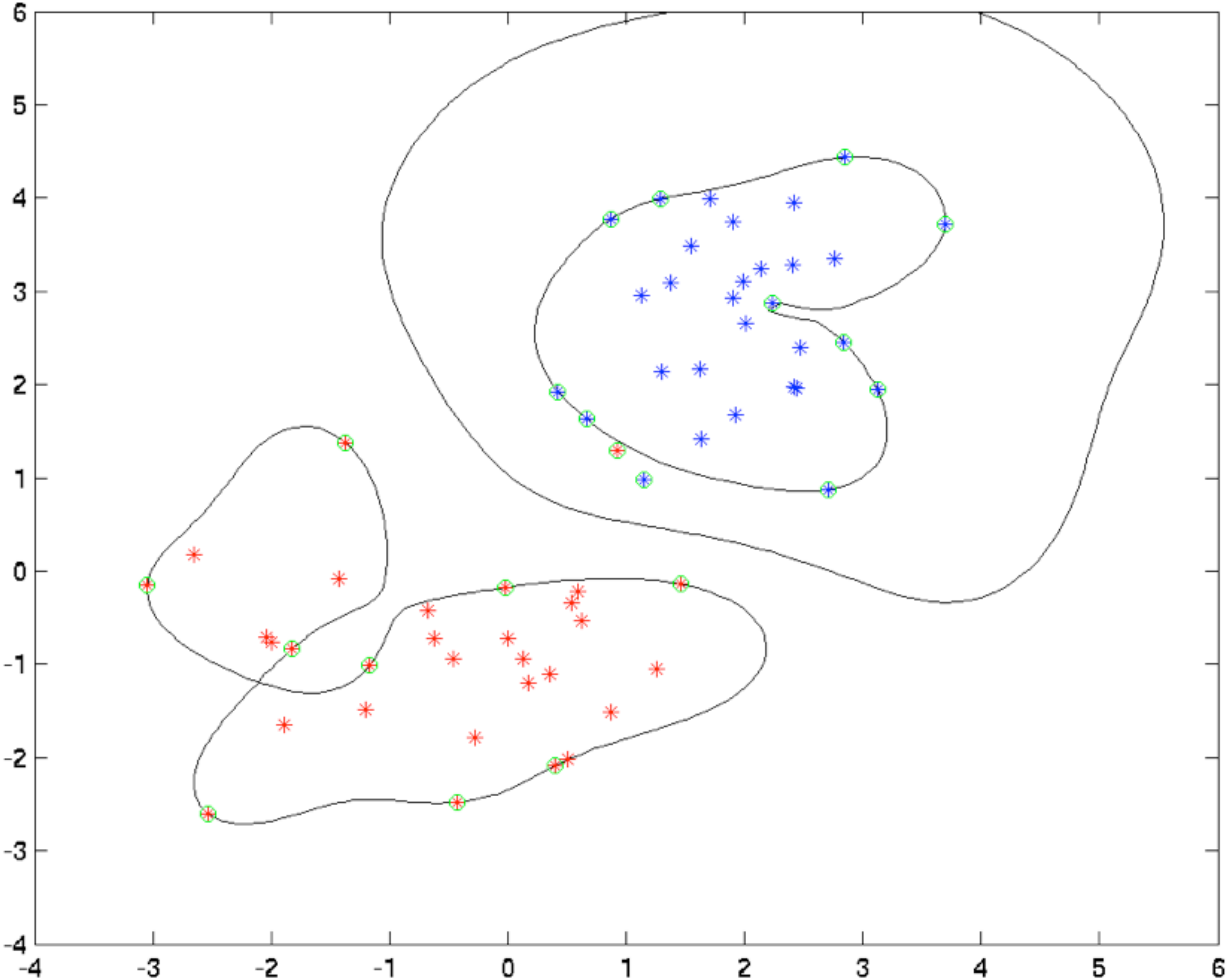
And now with a narrower kernel

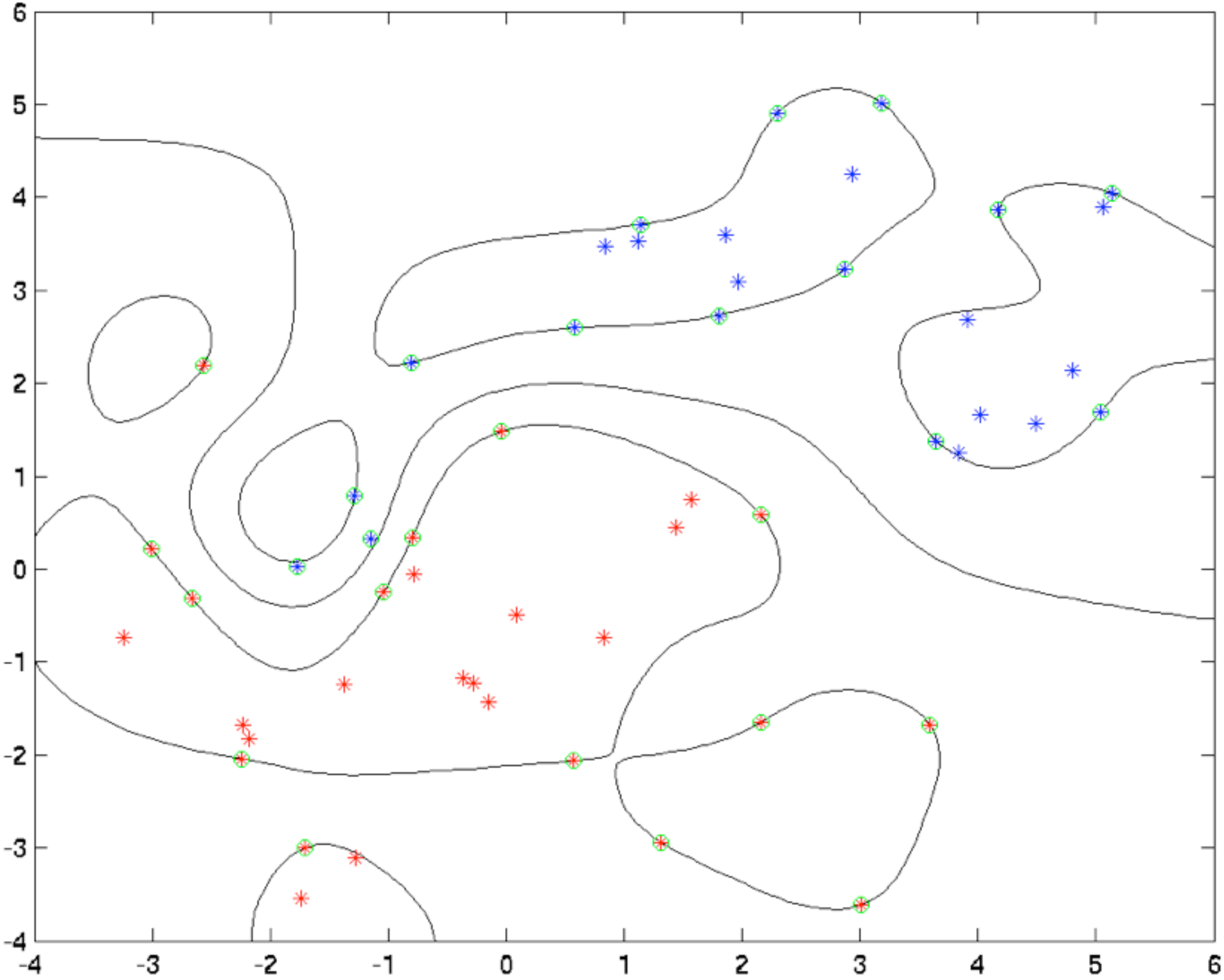


6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6



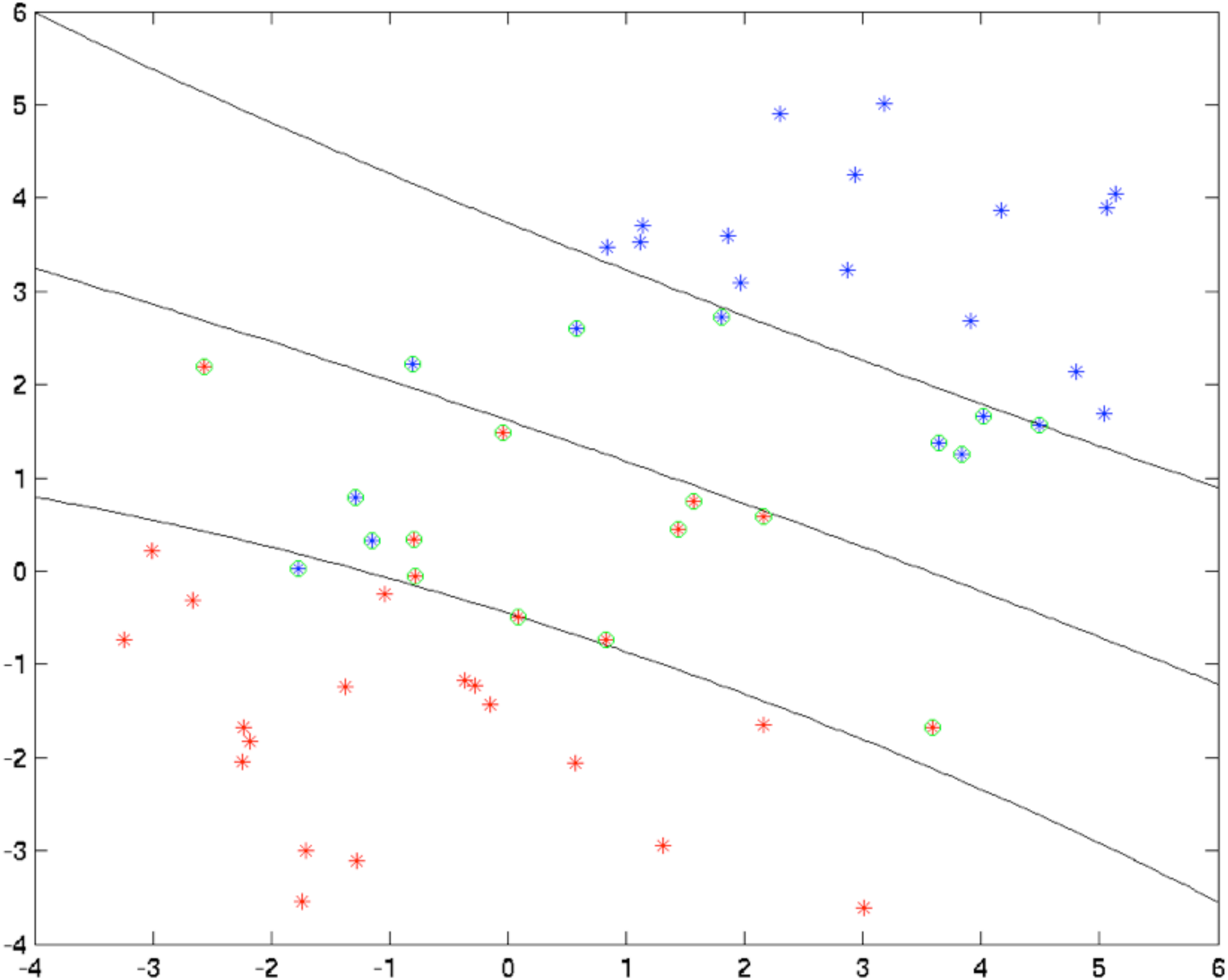
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6


6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6



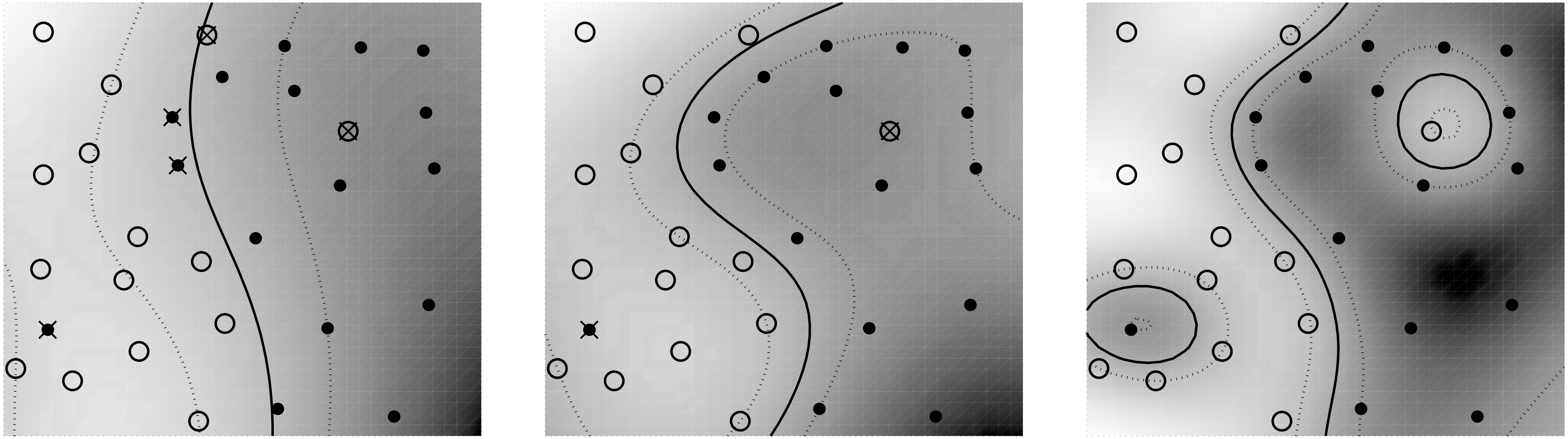
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

And now with a very wide kernel



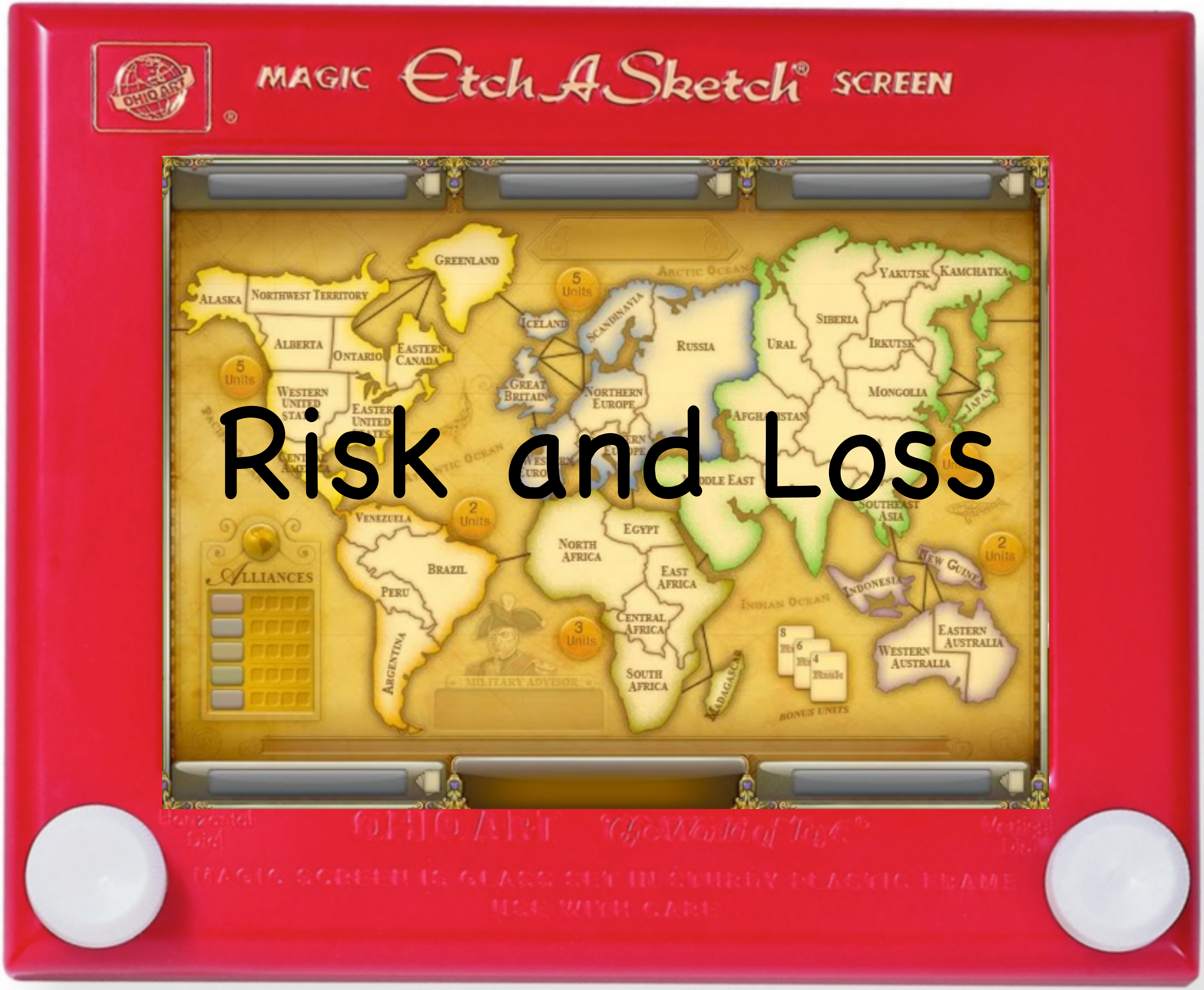
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

Nonlinear separation



- Increasing C allows for more nonlinearities
- Decreases number of errors
- SV boundary need not be contiguous
- Kernel width adjusts function class

Risk and Loss



Loss function point of view

- **Constrained quadratic program**

$$\underset{w,b}{\text{minimize}} \quad \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$\text{subject to } y_i\left[\langle w, x_i\rangle + b\right] \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

- **Risk minimization setting**

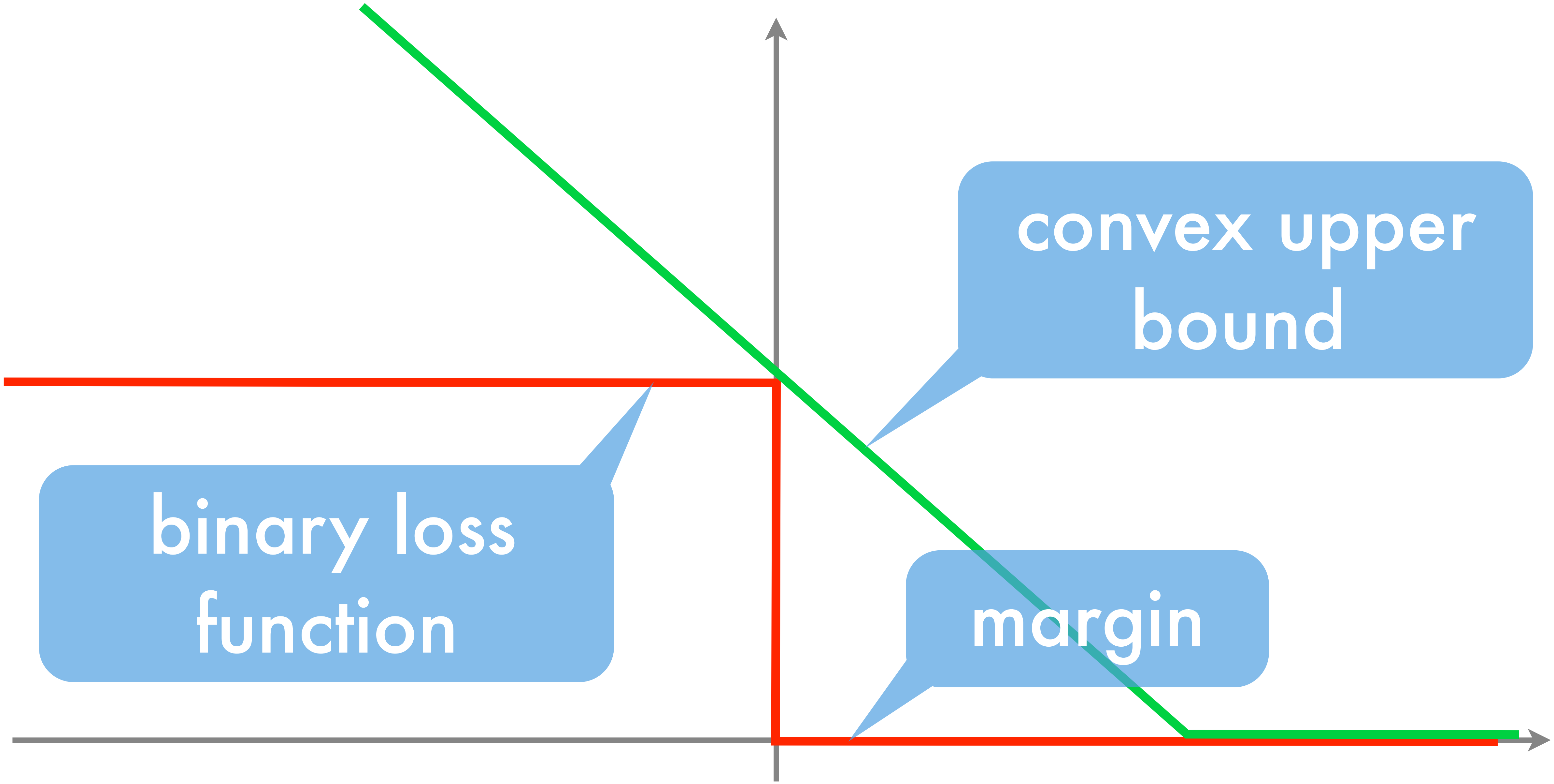
$$\underset{w,b}{\text{minimize}} \quad \frac{1}{2}\|w\|^2 + C\sum_i \max\left[0, 1 - y_i\left[\langle w, x_i\rangle + b\right]\right]$$

empirical risk

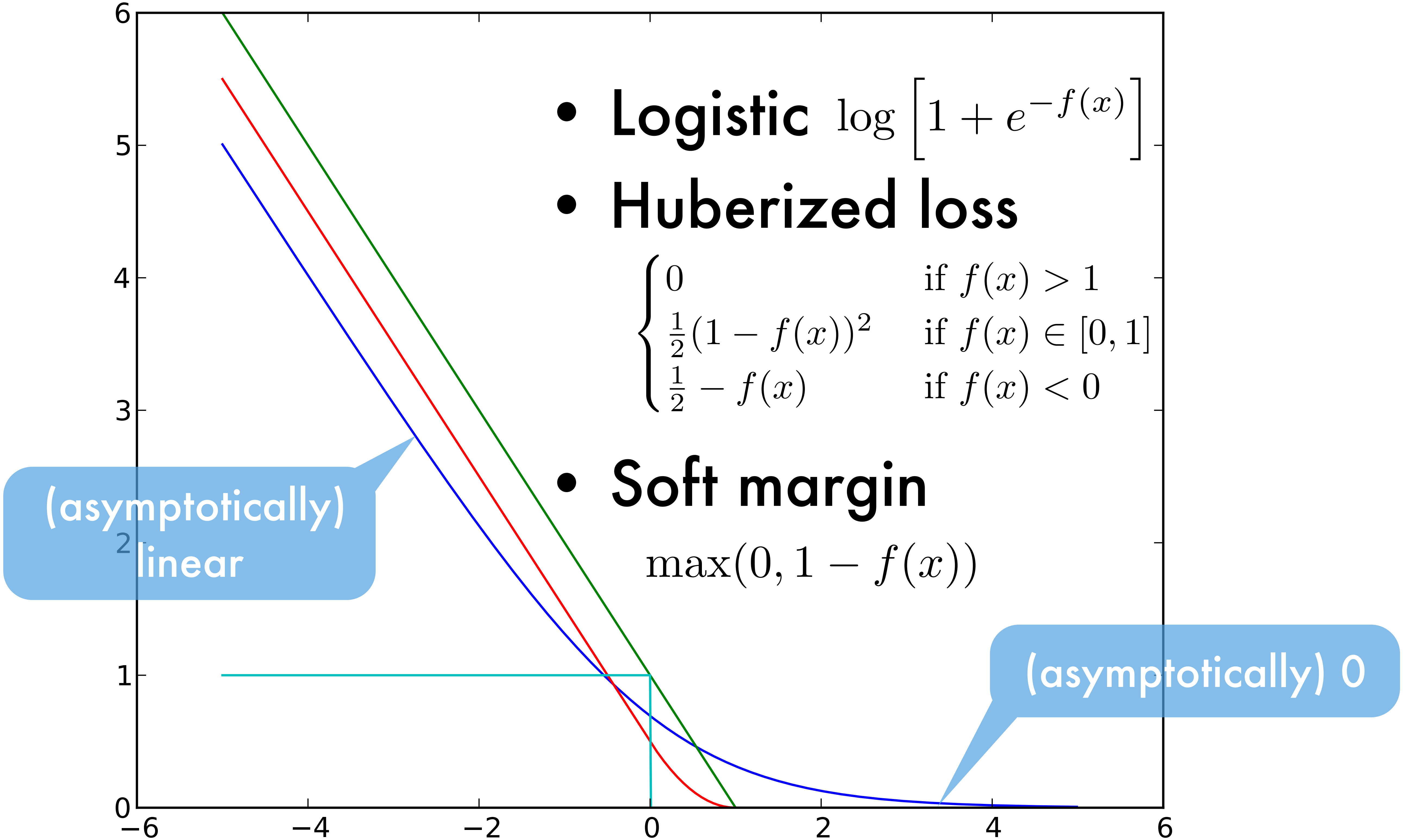
Follows from finding minimal slack variable for w

Soft margin as proxy for binary

- Soft margin loss $\max(0, 1 - yf(x))$
- Binary loss $\{yf(x) < 0\}$



More loss functions



- **Logistic** $\log\left[1 + e^{-f(x)}\right]$
- **Huberized loss**

$$\begin{cases} 0 & \text{if } f(x) > 1 \\ \frac{1}{2}(1 - f(x))^2 & \text{if } f(x) \in [0, 1] \\ \frac{1}{2} - f(x) & \text{if } f(x) < 0 \end{cases}$$

- **Soft margin**

$$\max(0, 1 - f(x))$$

Risk minimization view

- Find function f minimizing classification error

$$R[f] := \mathbf{E}_{x,y \sim p(x,y)} \left[\{ y f(x) > 0 \} \right]$$

- Compute empirical average

$$R_{\text{emp}}[f] := \frac{1}{m} \sum_{i=1}^{m} \{ y_i f(x_i) > 0 \}$$

  - Minimization is nonconvex
  - Overfitting as we minimize empirical error
- Compute convex upper bound on the loss
- Add regularization for capacity control

regularization

$$R_{\text{reg}}[f] := \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i f(x_i)) + \lambda \Omega[f]$$

how to control λ

Summary

- Support Vector Classification
  Large Margin Separation, optimization problem
- Properties
  Support Vectors, kernel expansion
- Soft margin classifier
  Dual problem, robustness