

BBM406

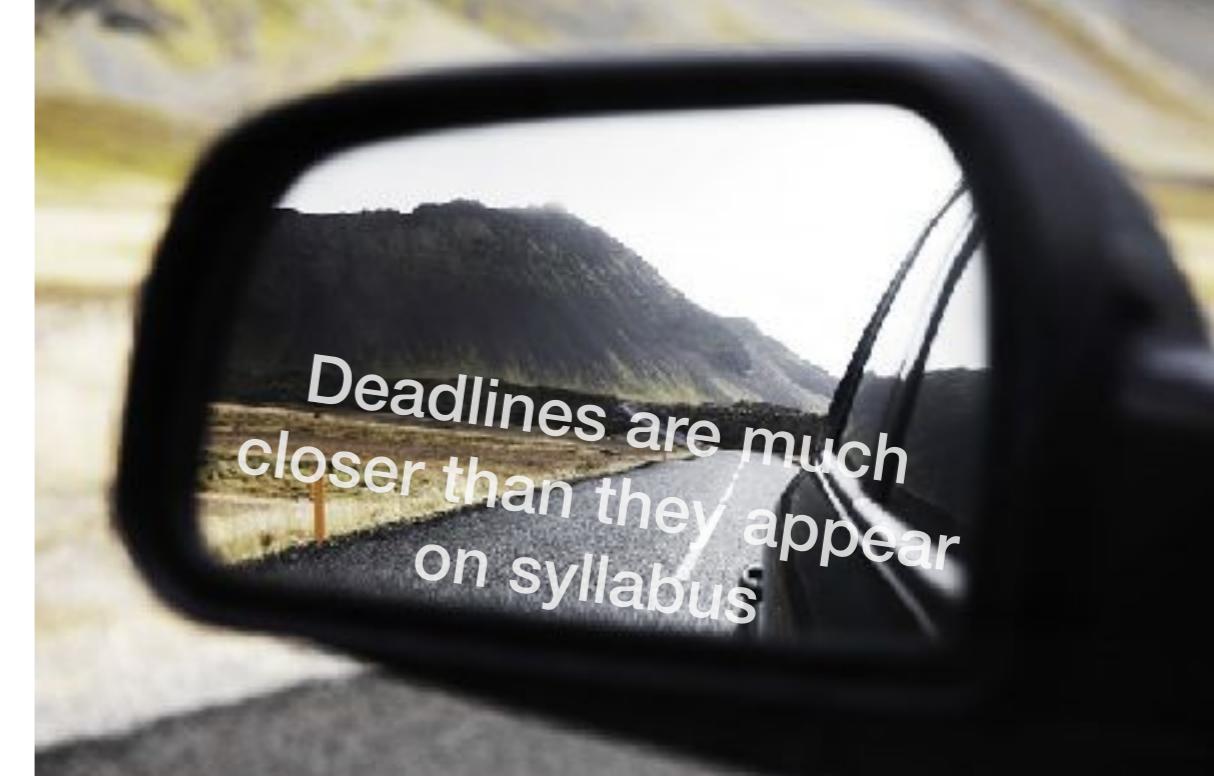
Fundamentals of Machine Learning

Lecture 17: Kernel Trick for SVMs Risk and Loss Support Vector Regression



Administrative

- Project progress reports are due soon!

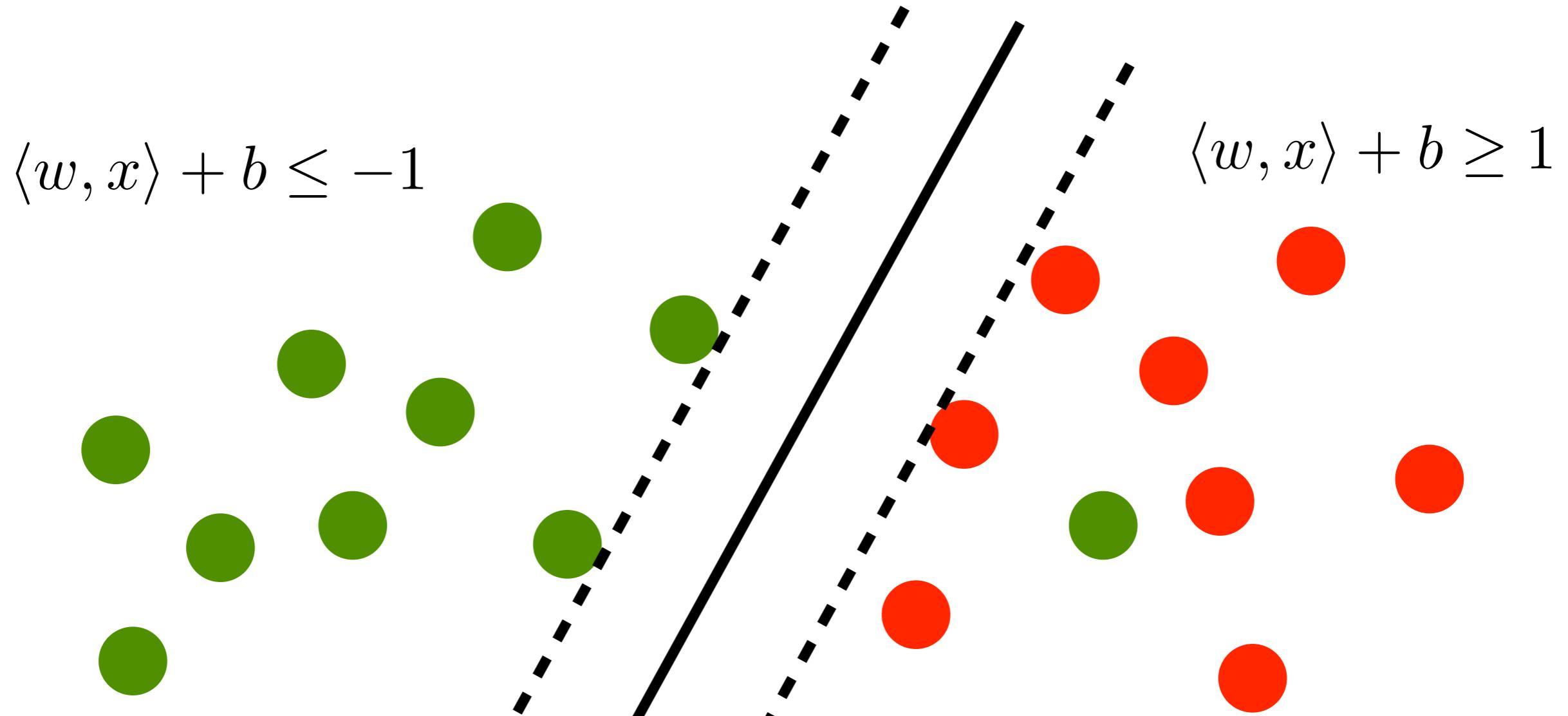


Due: December 22, 2019 (11:59pm)

Each group should submit a project progress report by December 22, 2018. The report should be **3-4 pages** and should describe the following points as clearly as possible:

- **Problem to be addressed.** Give a short description of the problem that you will explore. Explain why you find it interesting.
- **Related work.** Briefly review the major works related to your research topic.
- **Methodology to be employed.** Describe the neural architecture that is expected to form the basis of the project. State whether you will extend an existing method or you are going to devise your own approach.
- **Experimental evaluation.** Briefly explain how you will evaluate your results. State which dataset(s) you will employ in your evaluation. Provide your preliminary results (if any).

Last time... Soft-margin Classifier



minimum error separator
is impossible

Theorem (Minsky & Papert)

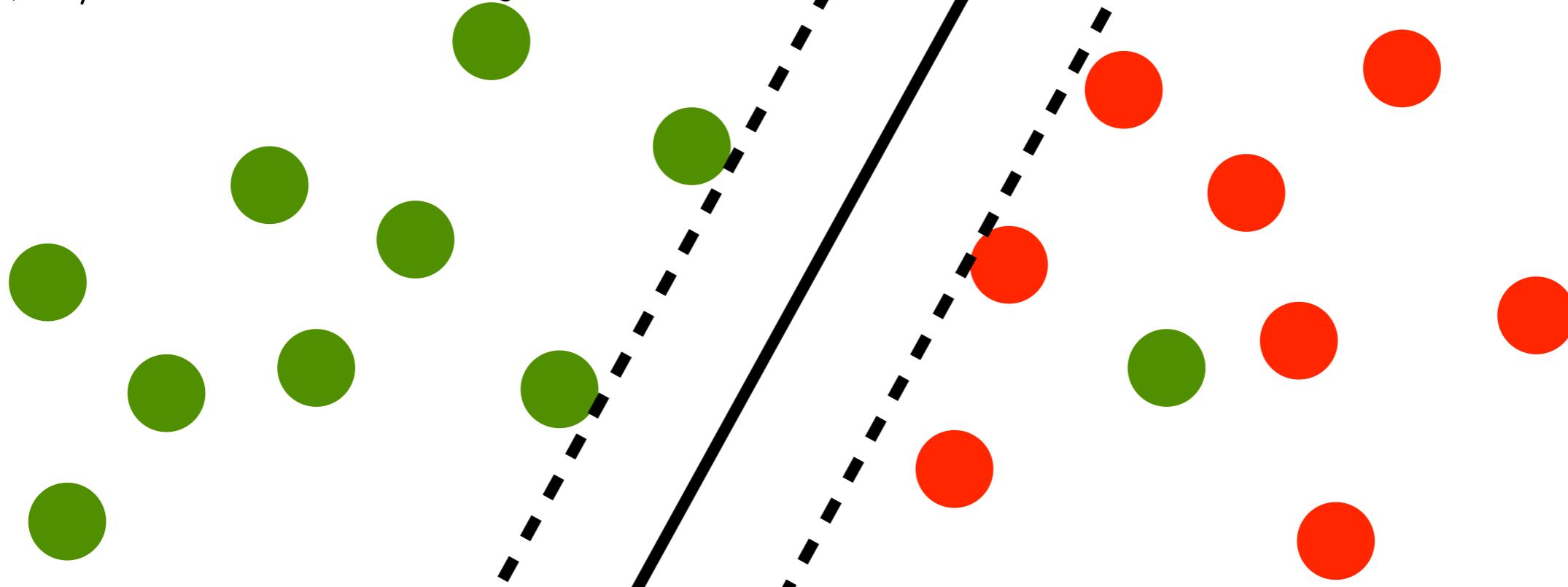
Finding the minimum error separating hyperplane is NP hard

Last time... Adding Slack Variables

$$\xi_i \geq 0$$

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

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

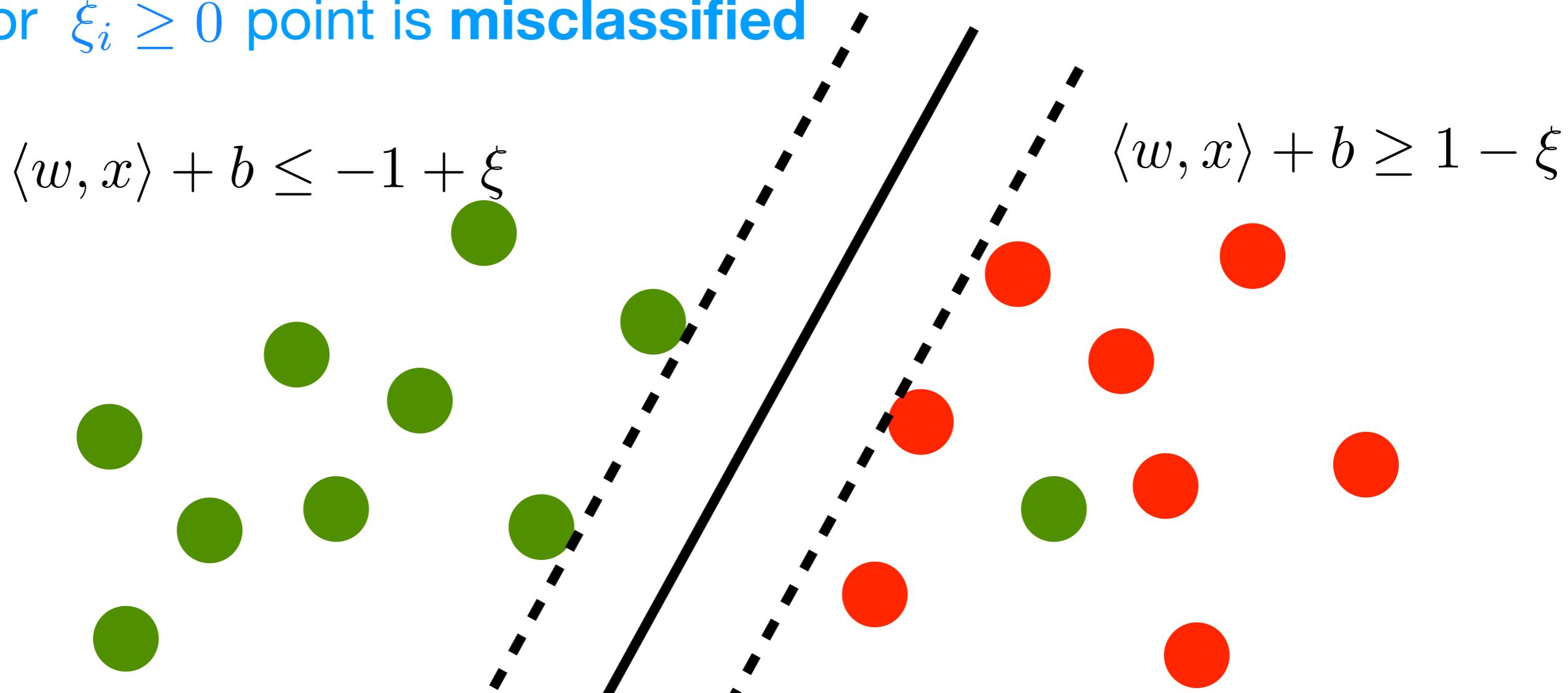


minimize amount
of slack

Convex optimization problem

Last time... Adding Slack Variables

- for $0 < \xi \leq 1$ point is between the margin and **correctly classified**
- for $\xi_i \geq 0$ point is **misclassified**



minimize amount
of slack

Convex optimization problem

Last time... Adding Slack Variables

- Hard margin problem

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

- With slack variables

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

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

Problem is always feasible. Proof:

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

Soft-margin classifier

- Optimization problem:

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

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

C is a regularization parameter:

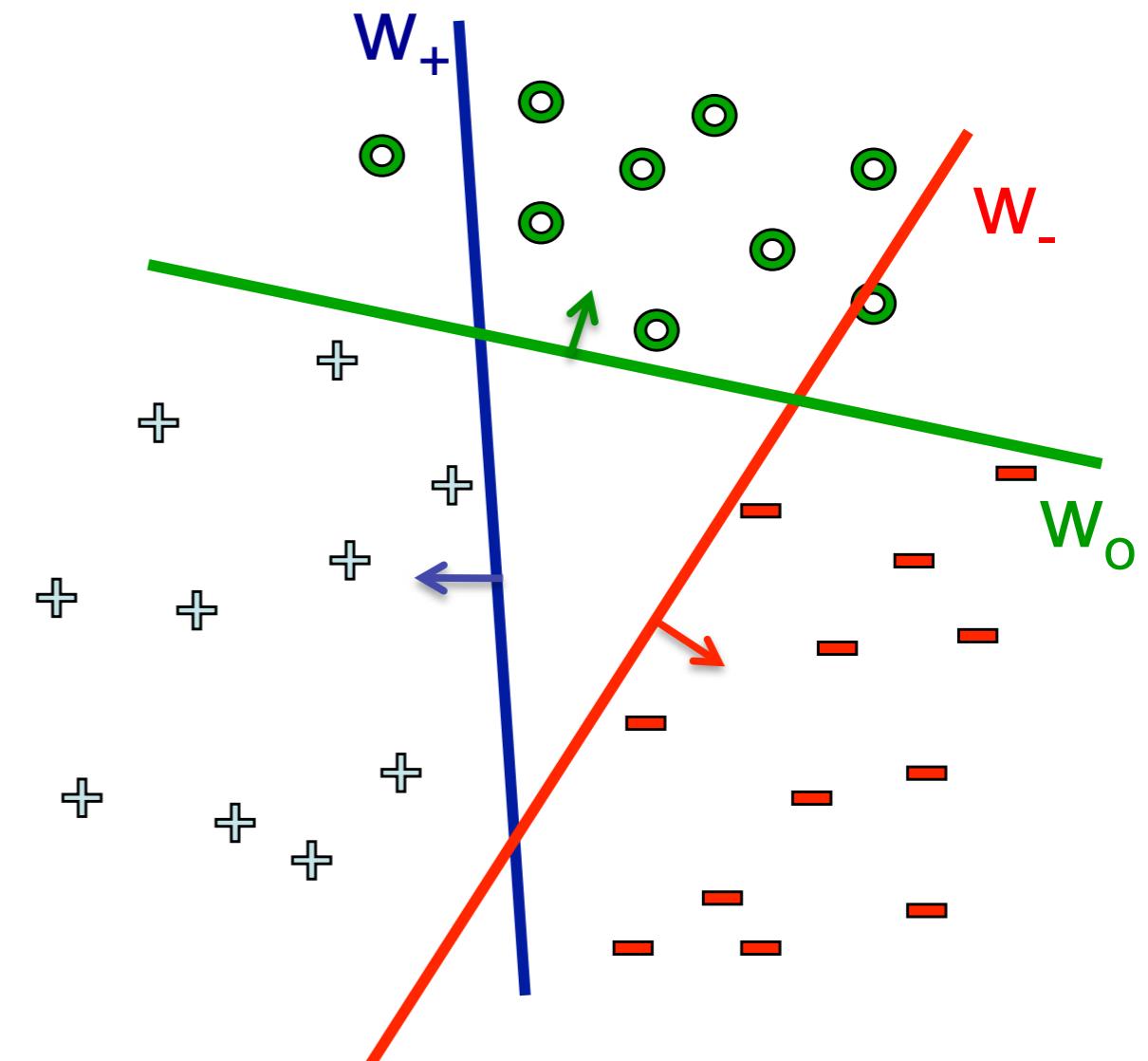
- small C allows constraints to be easily ignored
→ **large margin**
- large C makes constraints hard to ignore
→ **narrow margin**
- $C = \infty$ enforces all constraints: **hard margin**

Last time... Multi-class SVM

- Simultaneously learn 3 sets of weights:
- How do we guarantee the correct labels?
- Need new constraints!

The “score” of the correct class must be better than the “score” of wrong classes:

$$w^{(y_j)} \cdot x_j + b^{(y_j)} > w^{(y)} \cdot x_j + b^{(y)} \quad \forall j, y \neq y_j$$



Last time... Multi-class SVM

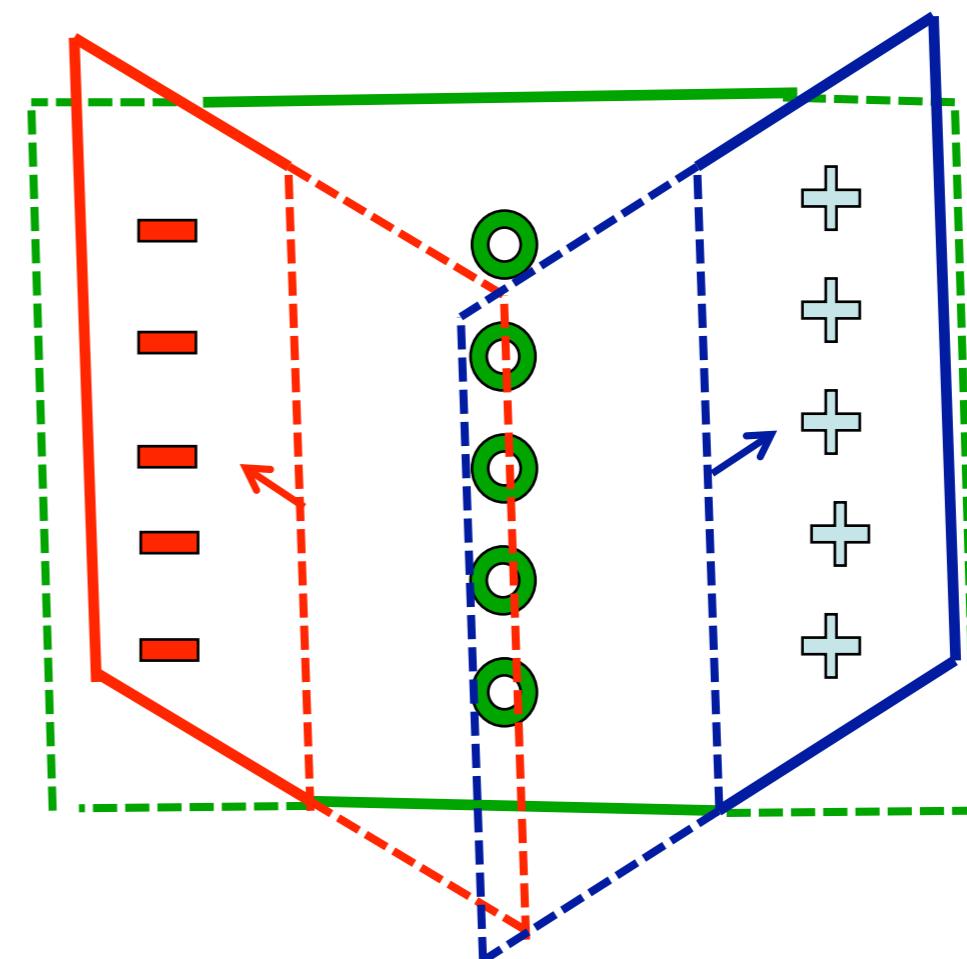
- As for the SVM, we introduce slack variables and maximize margin:

$$\begin{aligned} \text{minimize}_{\mathbf{w}, b} \quad & \sum_y \mathbf{w}^{(y)} \cdot \mathbf{w}^{(y)} + C \sum_j \xi_j \\ \mathbf{w}^{(y_j)} \cdot \mathbf{x}_j + b^{(y_j)} \geq & \mathbf{w}^{(y')} \cdot \mathbf{x}_j + b^{(y')} + 1 - \xi_j, \quad \forall y' \neq y_j, \quad \forall j \\ \xi_j \geq 0, \quad & \forall j \end{aligned}$$

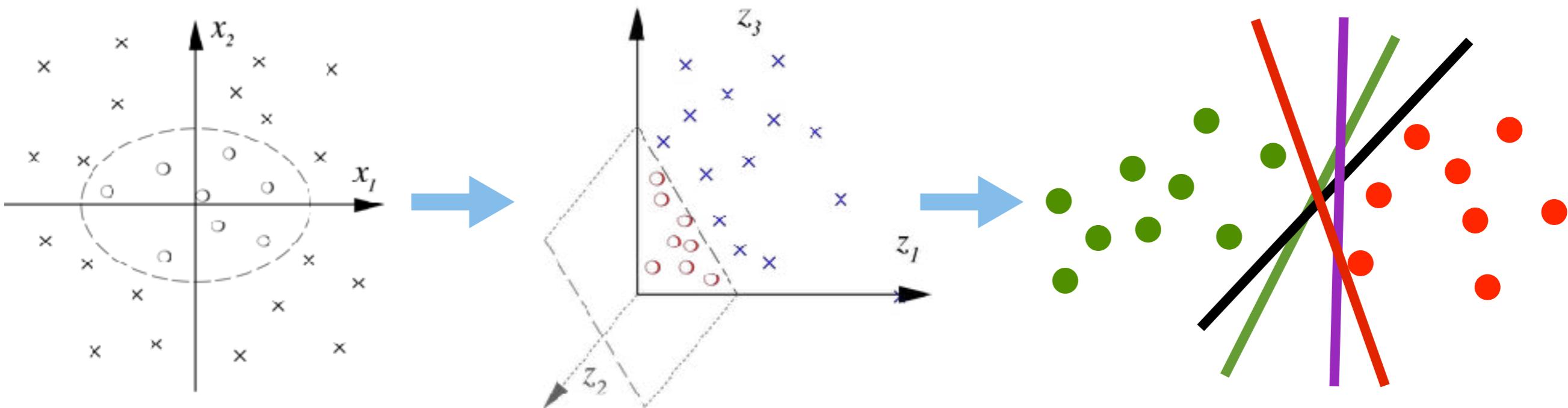
To predict, we use:

$$\hat{y} \leftarrow \arg \max_k w_k \cdot x + b_k$$

Now can we learn it? →



Last time... Kernels



- Original data
- Data in feature space (implicit)
- Solve in feature space using kernels

Last time... Quadratic Features

Quadratic Features in \mathbb{R}^2

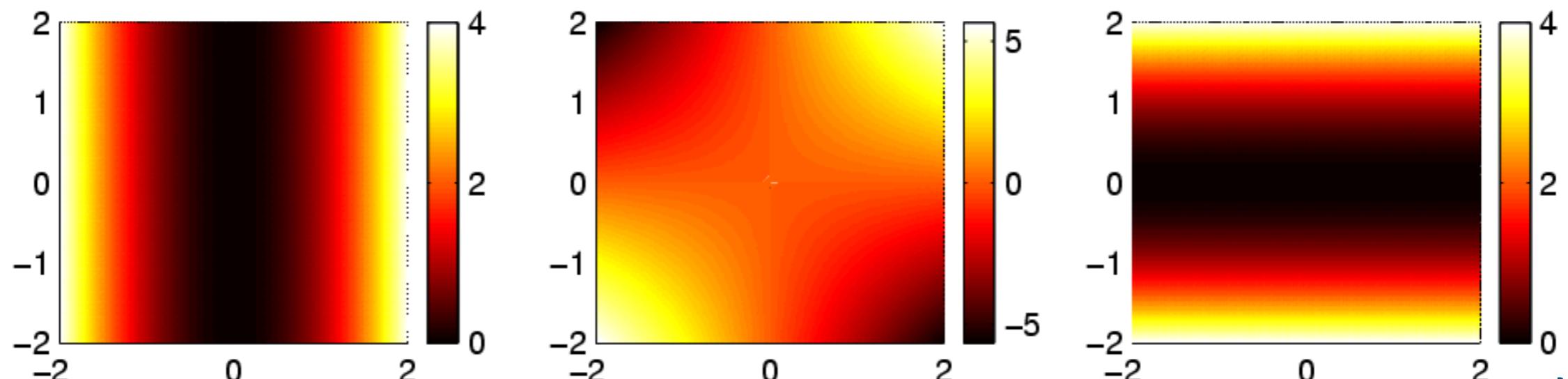
$$\Phi(x) := \left(x_1^2, \sqrt{2}x_1x_2, x_2^2 \right)$$

Dot Product

$$\begin{aligned} \langle \Phi(x), \Phi(x') \rangle &= \left\langle \left(x_1^2, \sqrt{2}x_1x_2, x_2^2 \right), \left({x'_1}^2, \sqrt{2}x'_1x'_2, {x'_2}^2 \right) \right\rangle \\ &= \langle x, x' \rangle^2. \end{aligned}$$

Insight

Trick works for any polynomials of order via $\langle x, x' \rangle^d$.



Computational Efficiency

Problem

- Extracting features can sometimes be very costly.
- Example: second order features in 1000 dimensions.
This leads to $5 \cdot 10^5$ numbers. For higher order polynomial features much worse.

Solution

Don't compute the features, try to compute dot products implicitly. For some features this works . . .

Definition

A kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a symmetric function in its arguments for which the following property holds

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle \text{ for some feature map } \Phi.$$

If $k(x, x')$ is much cheaper to compute than $\Phi(x)$. . .

Last time.. Example kernels

Examples of kernels $k(x, x')$

Linear

$$\langle x, x' \rangle$$

Laplacian RBF

$$\exp(-\lambda \|x - x'\|)$$

Gaussian RBF

$$\exp(-\lambda \|x - x'\|^2)$$

Polynomial

$$(\langle x, x' \rangle + c)^d, c \geq 0, d \in \mathbb{N}$$

B-Spline

$$B_{2n+1}(x - x')$$

Cond. Expectation

$$\mathbf{E}_c[p(x|c)p(x'|c)]$$

Simple trick for checking Mercer's condition

Compute the Fourier transform of the kernel and check that it is nonnegative.

Today

- The Kernel Trick for SVMs
- Risk and Loss
- Support Vector Regression

The Kernel Trick for SVMs

The Kernel Trick for SVMs

- Linear soft margin problem

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

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

- Dual problem

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

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

- Support vector expansion

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

The Kernel Trick for SVMs

- Linear soft margin problem

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

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

- Dual problem

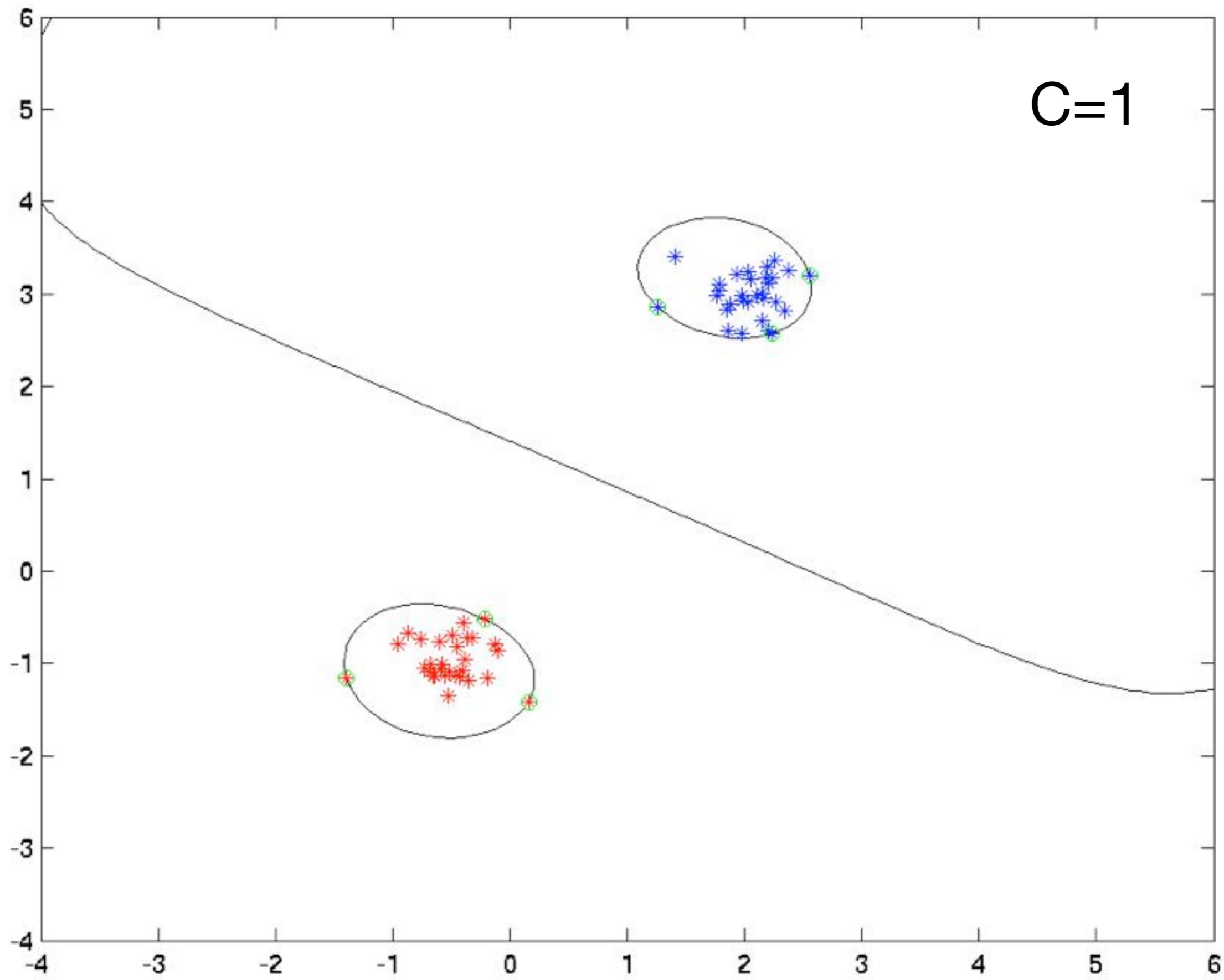
$$\underset{\alpha}{\text{maximize}} \quad -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) + \sum_i \alpha_i$$

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

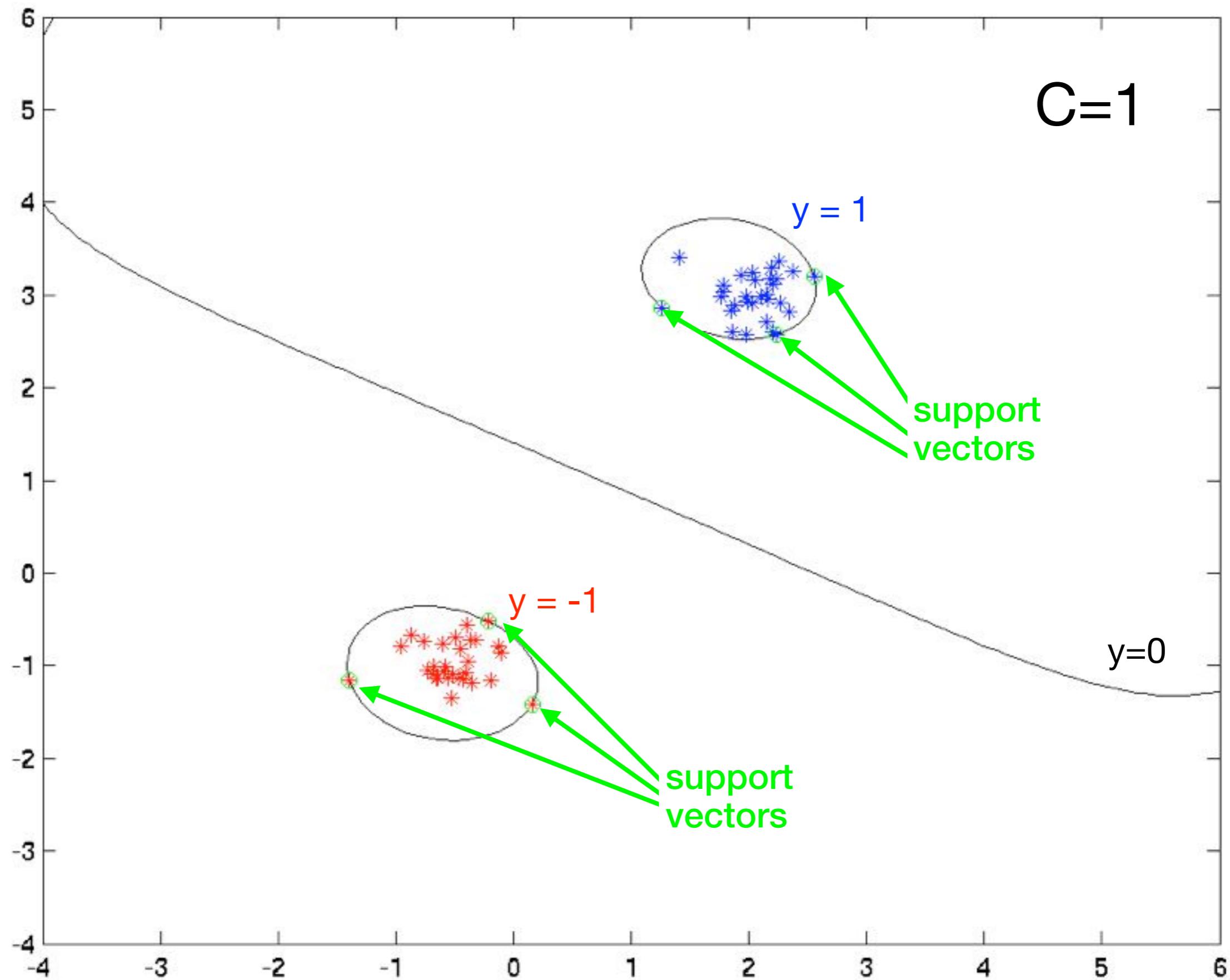
- Support vector expansion

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

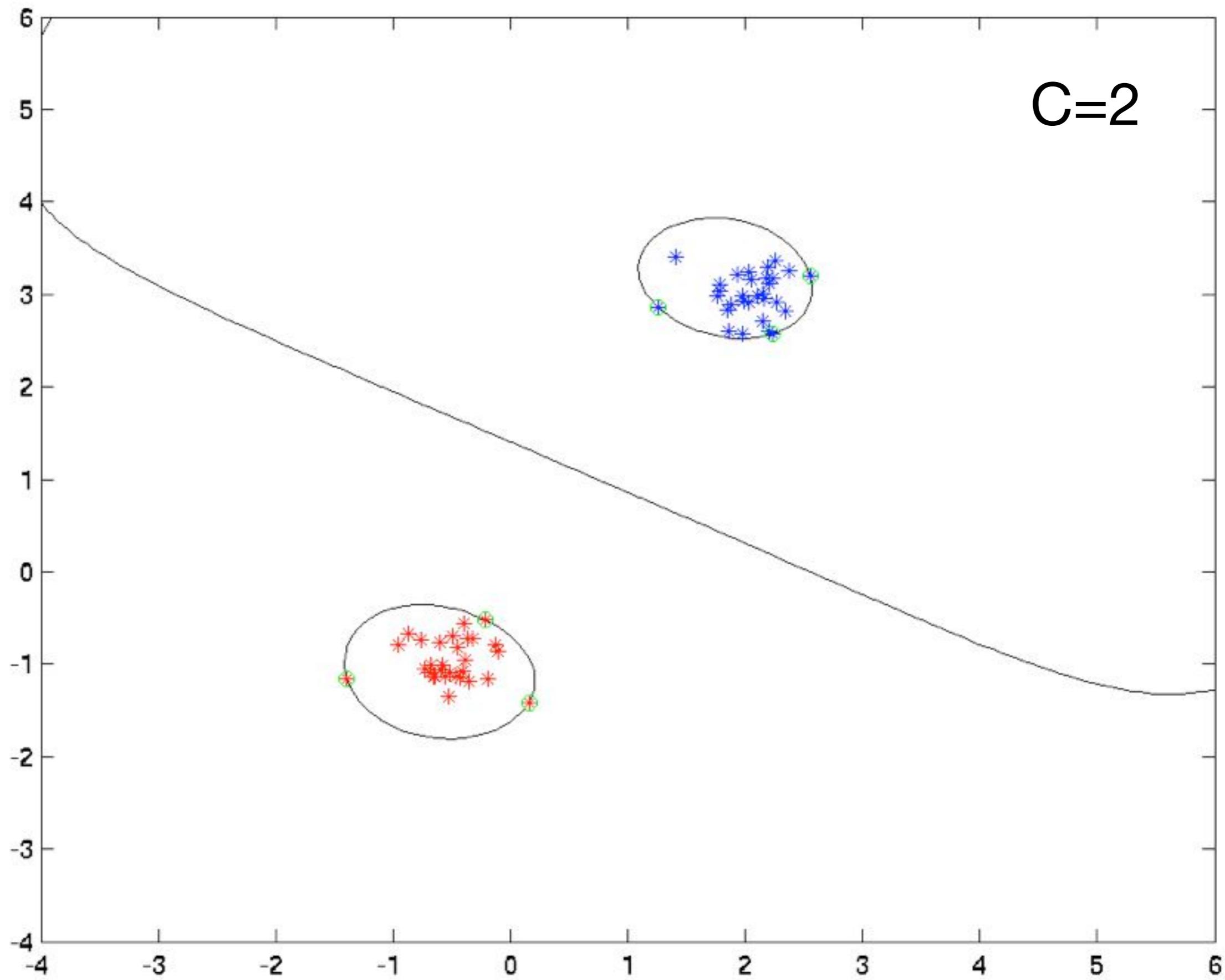
C=1



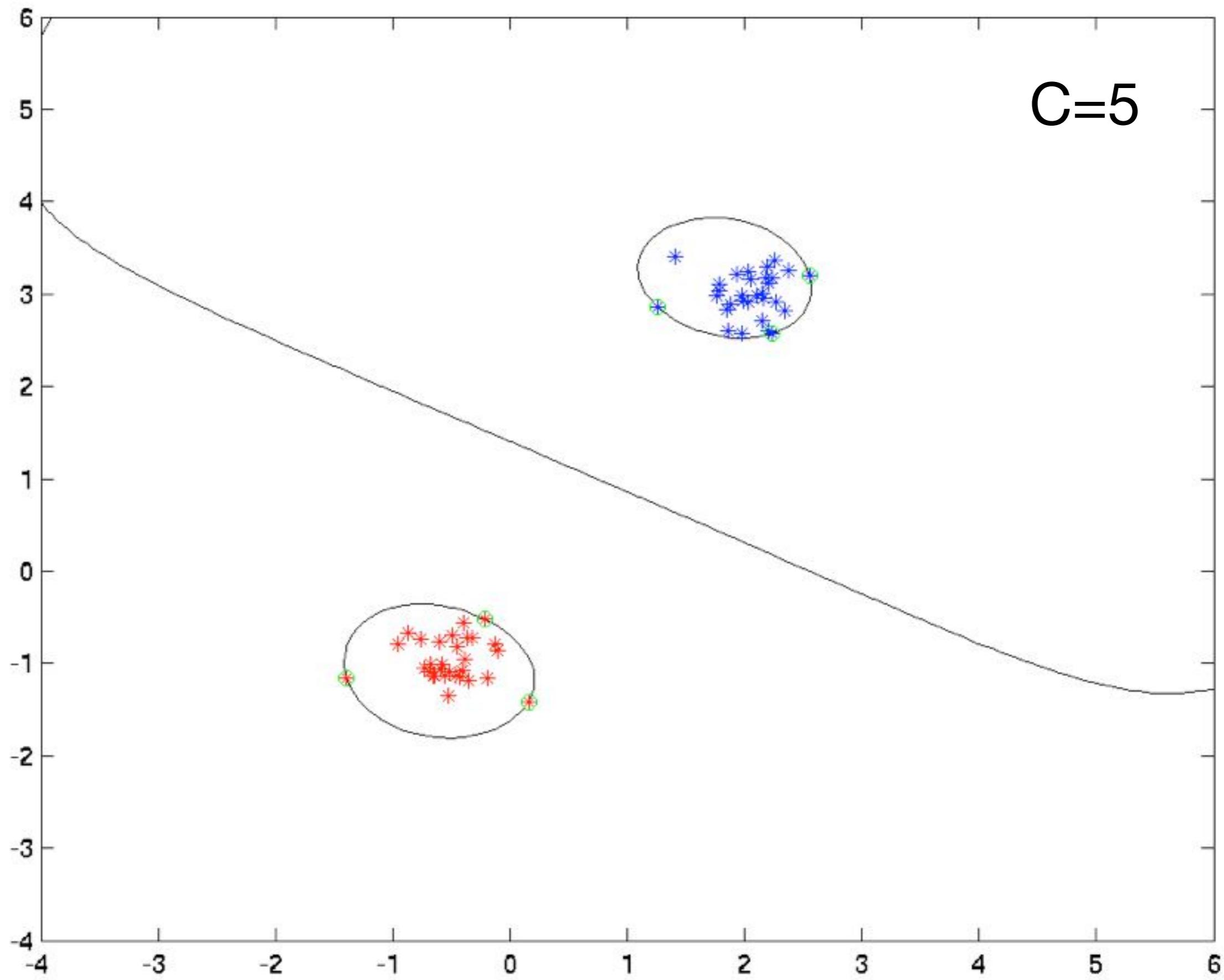
$C=1$



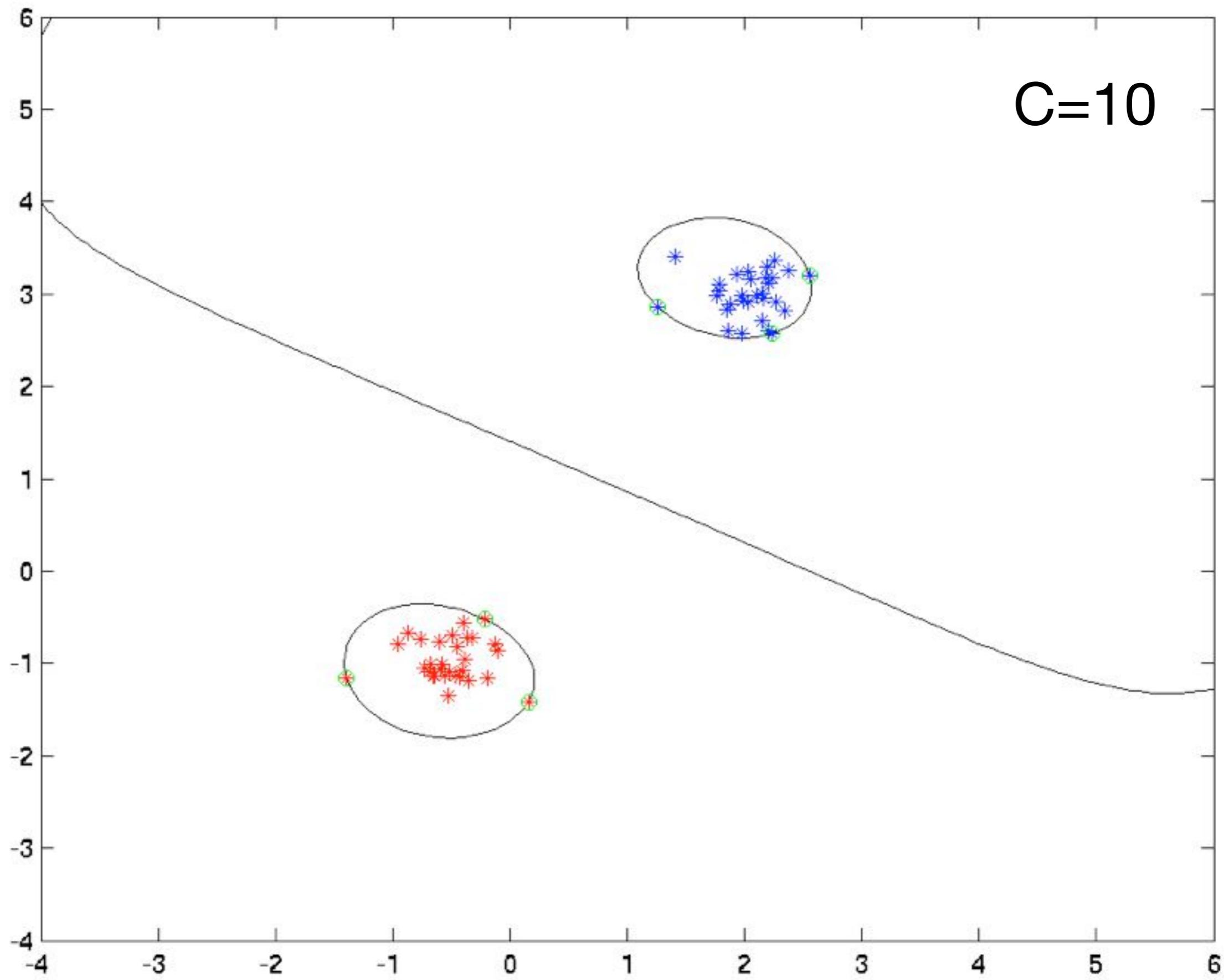
$C=2$



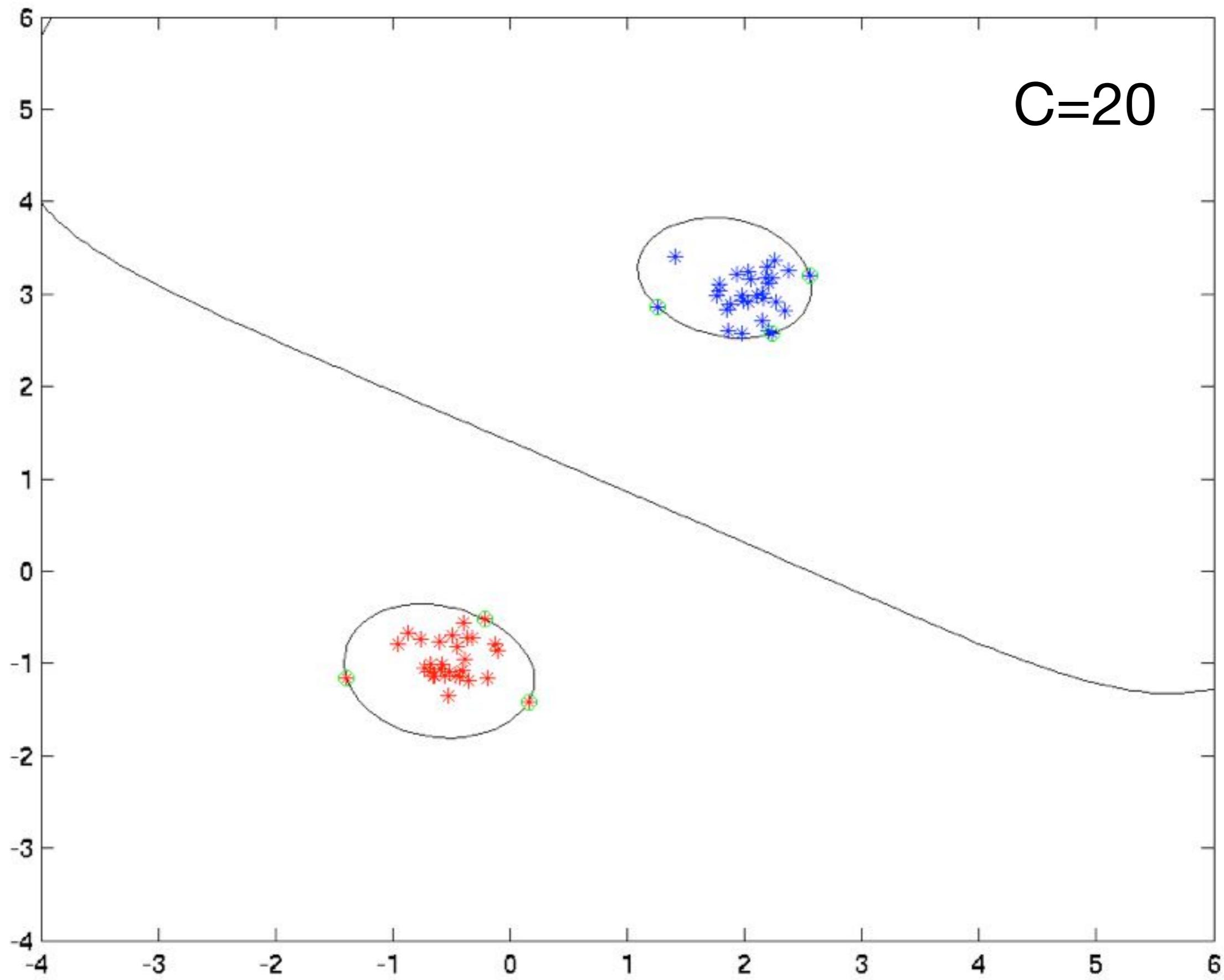
C=5



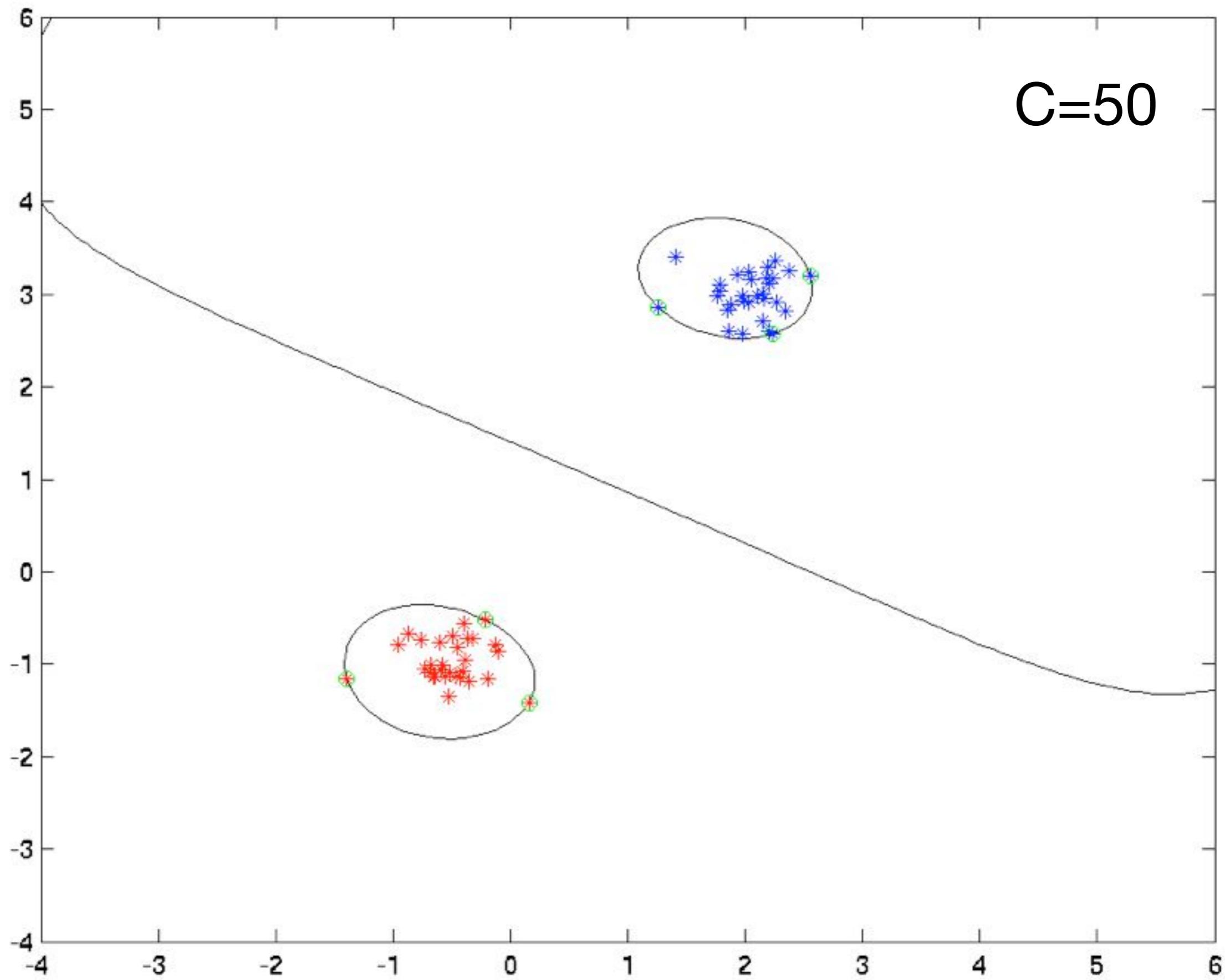
C=10



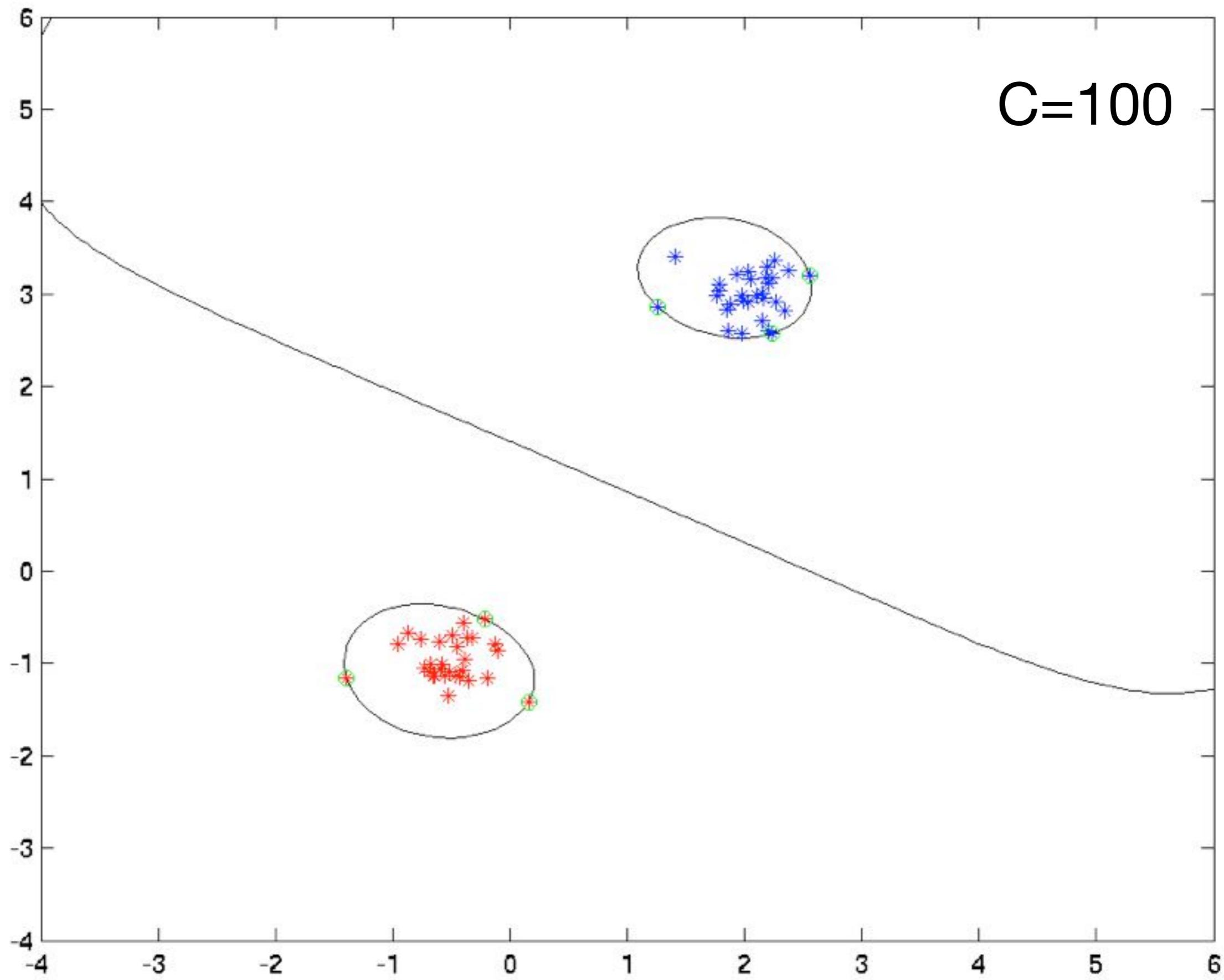
C=20



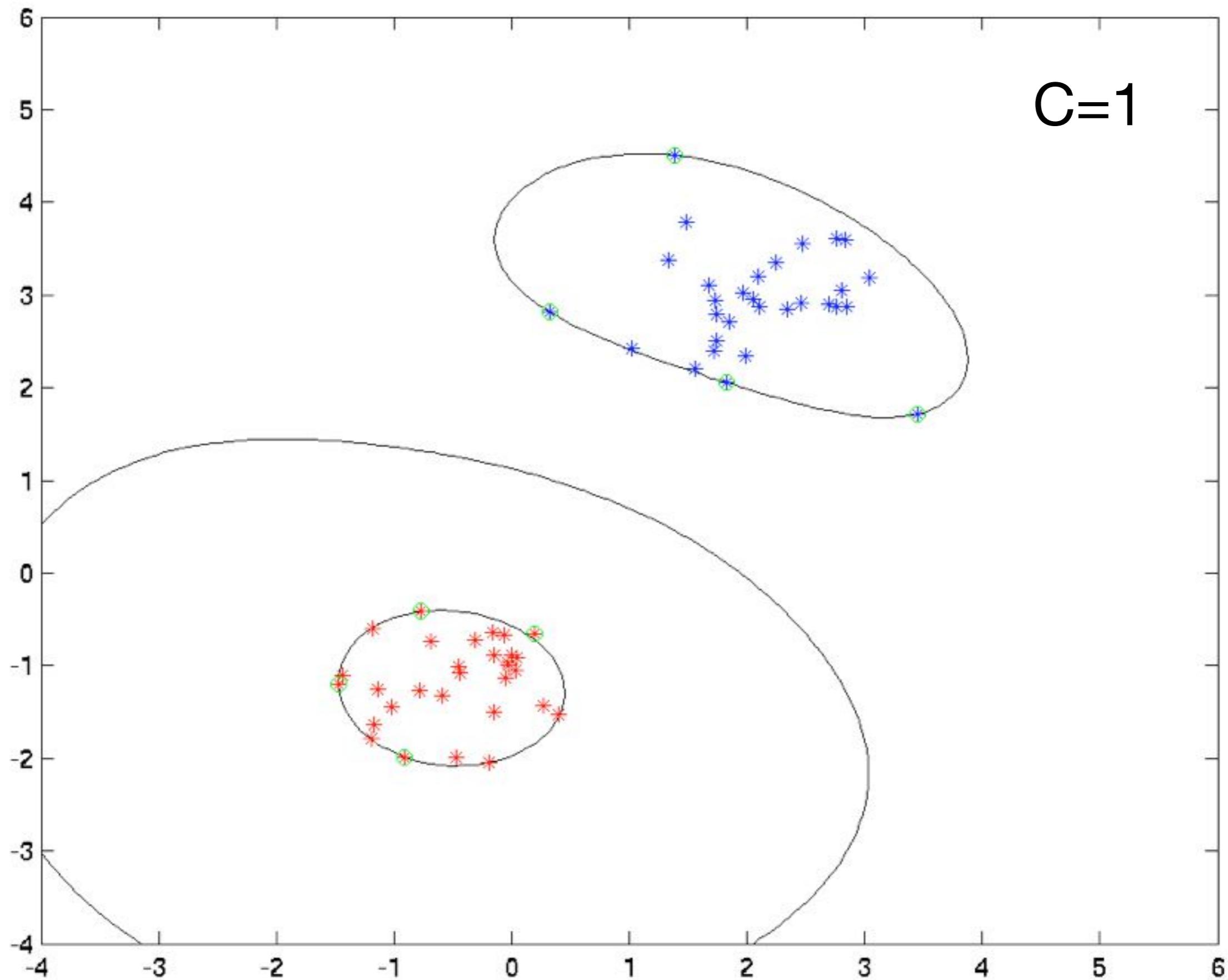
C=50



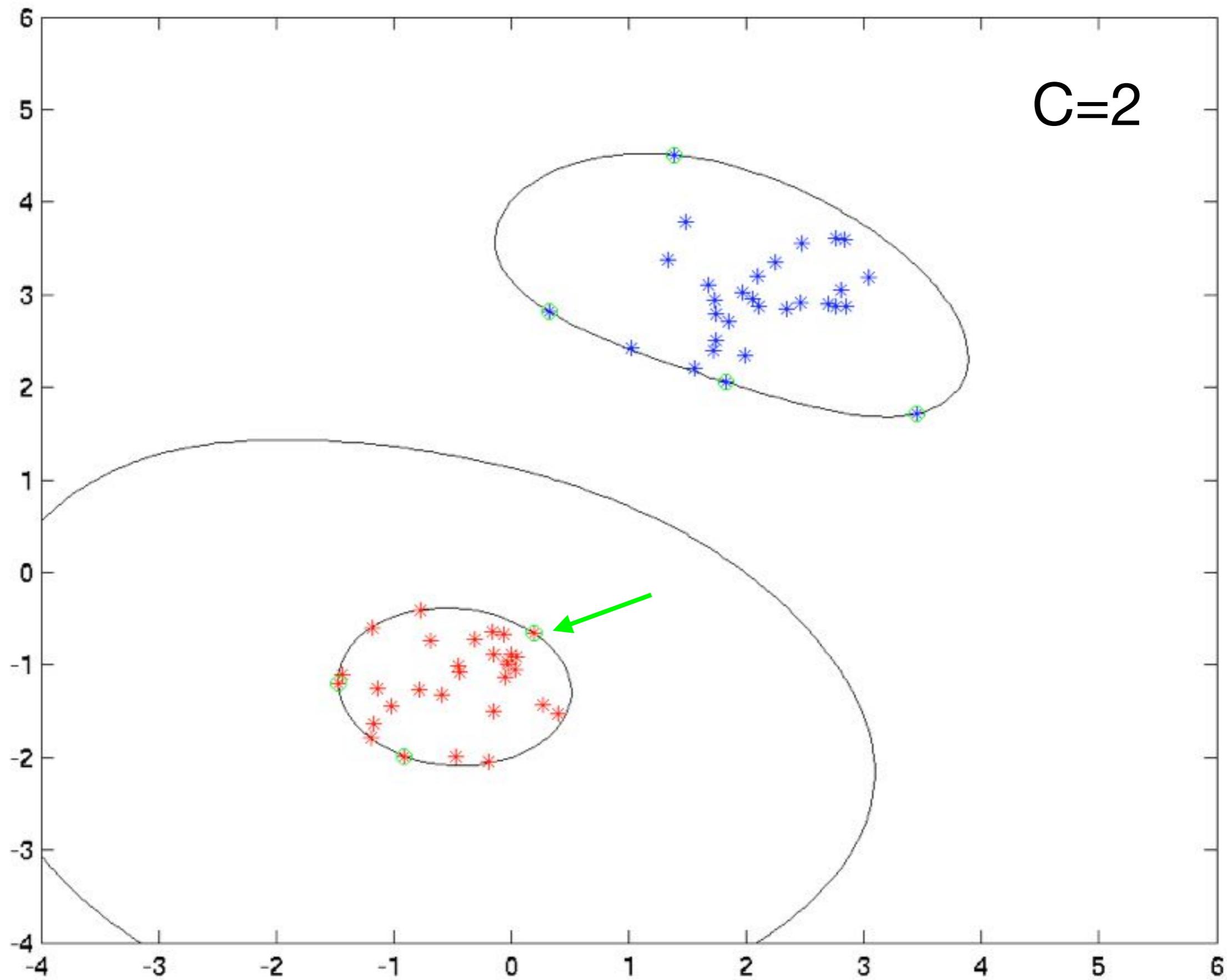
$C=100$



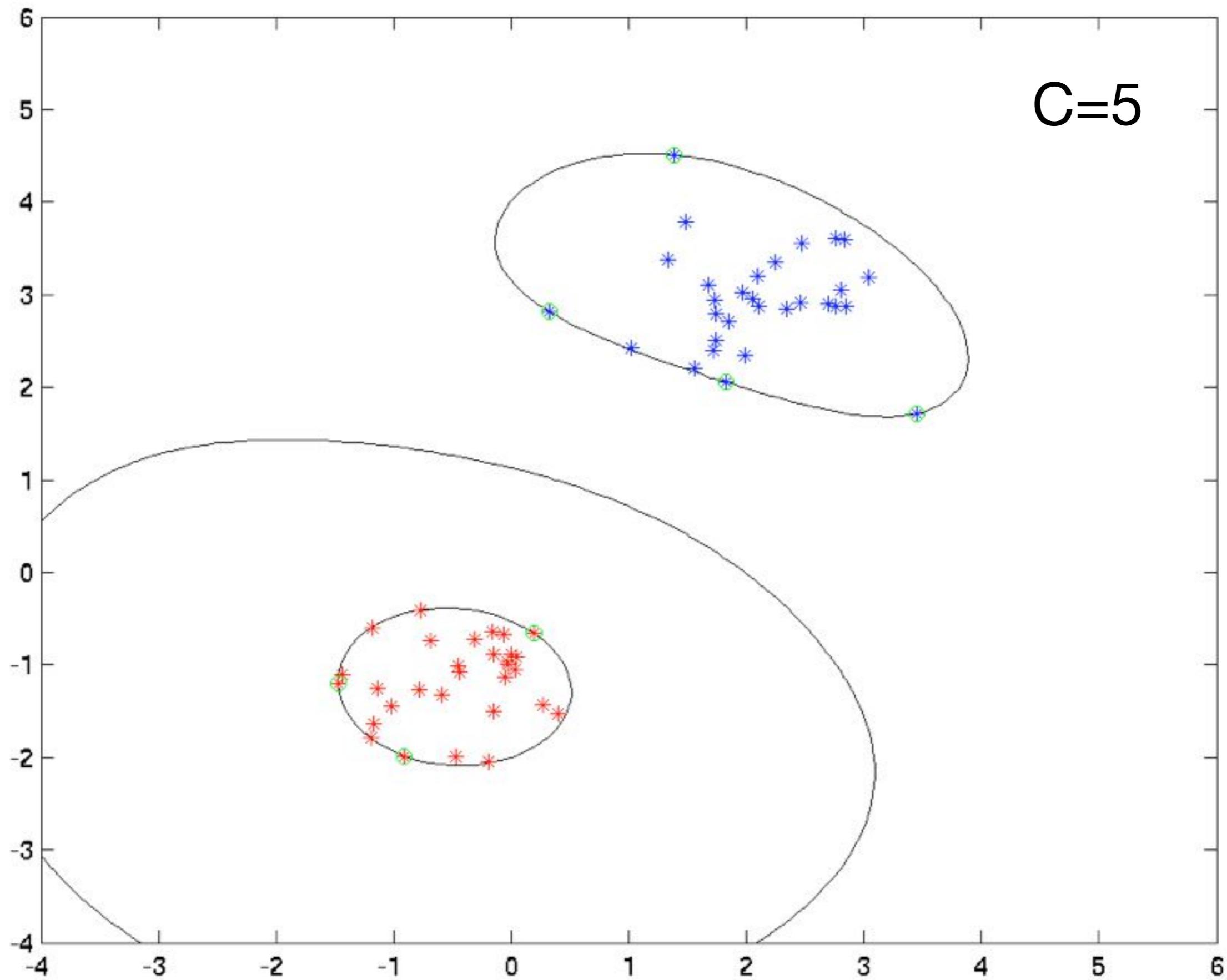
C=1



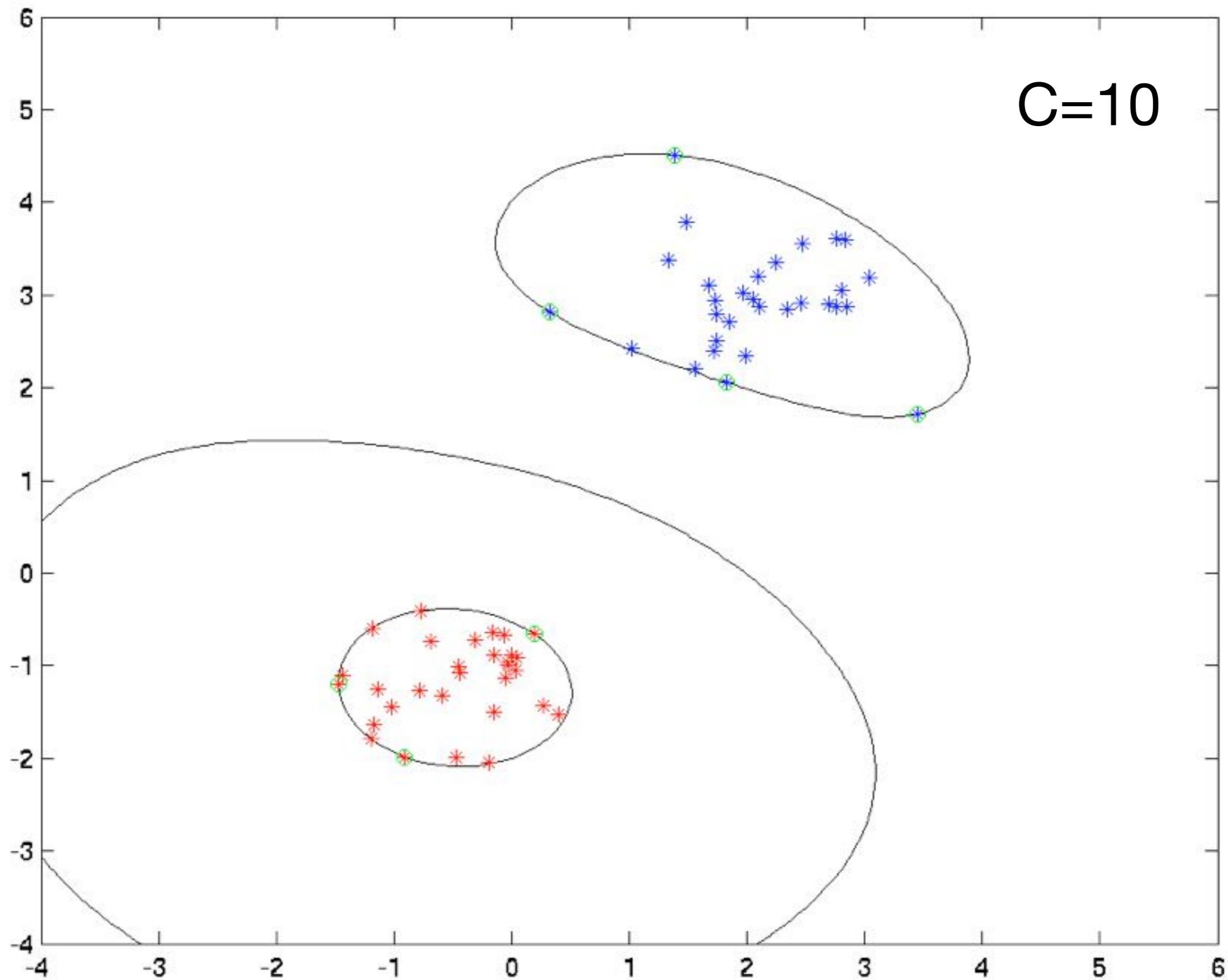
$C=2$



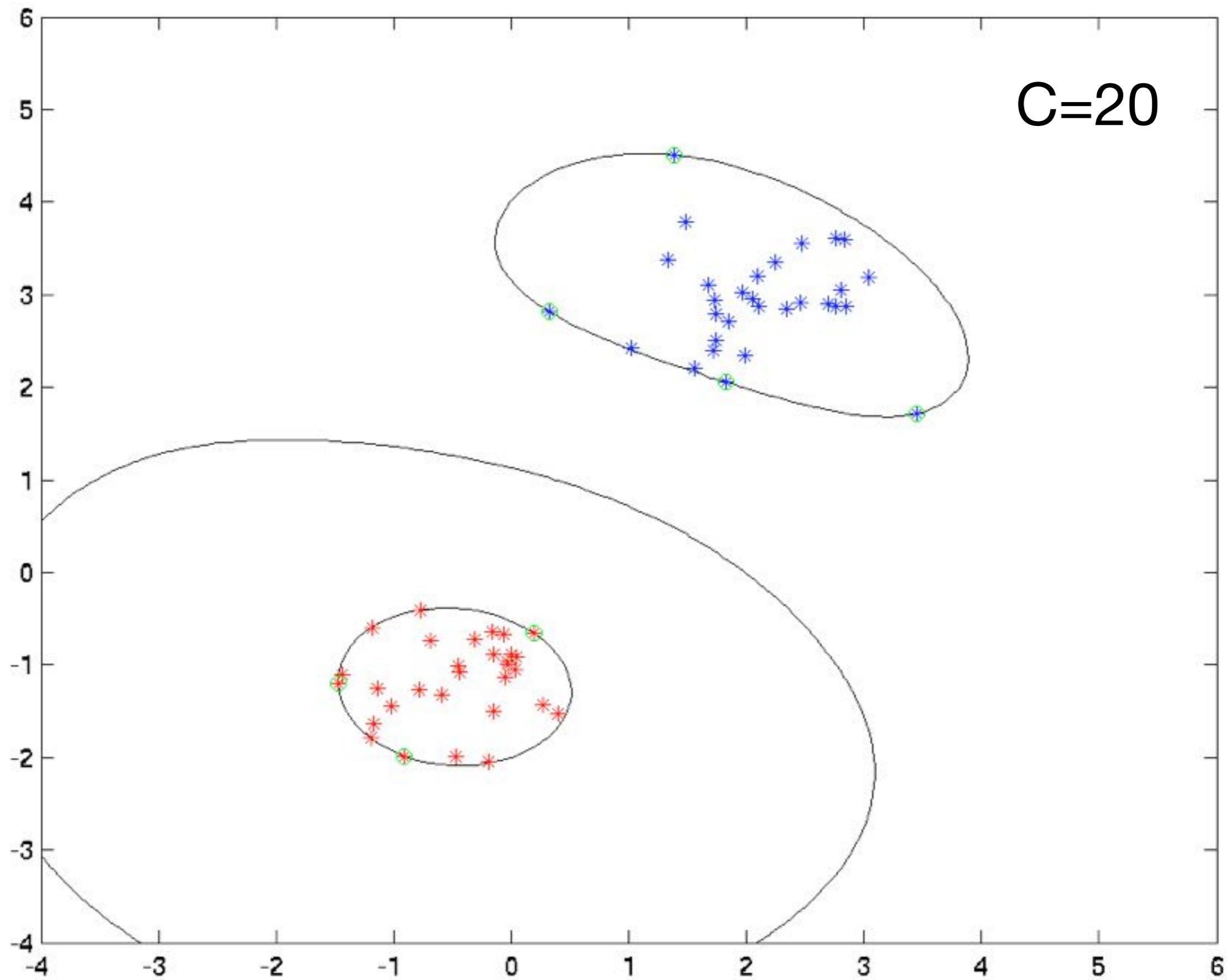
C=5



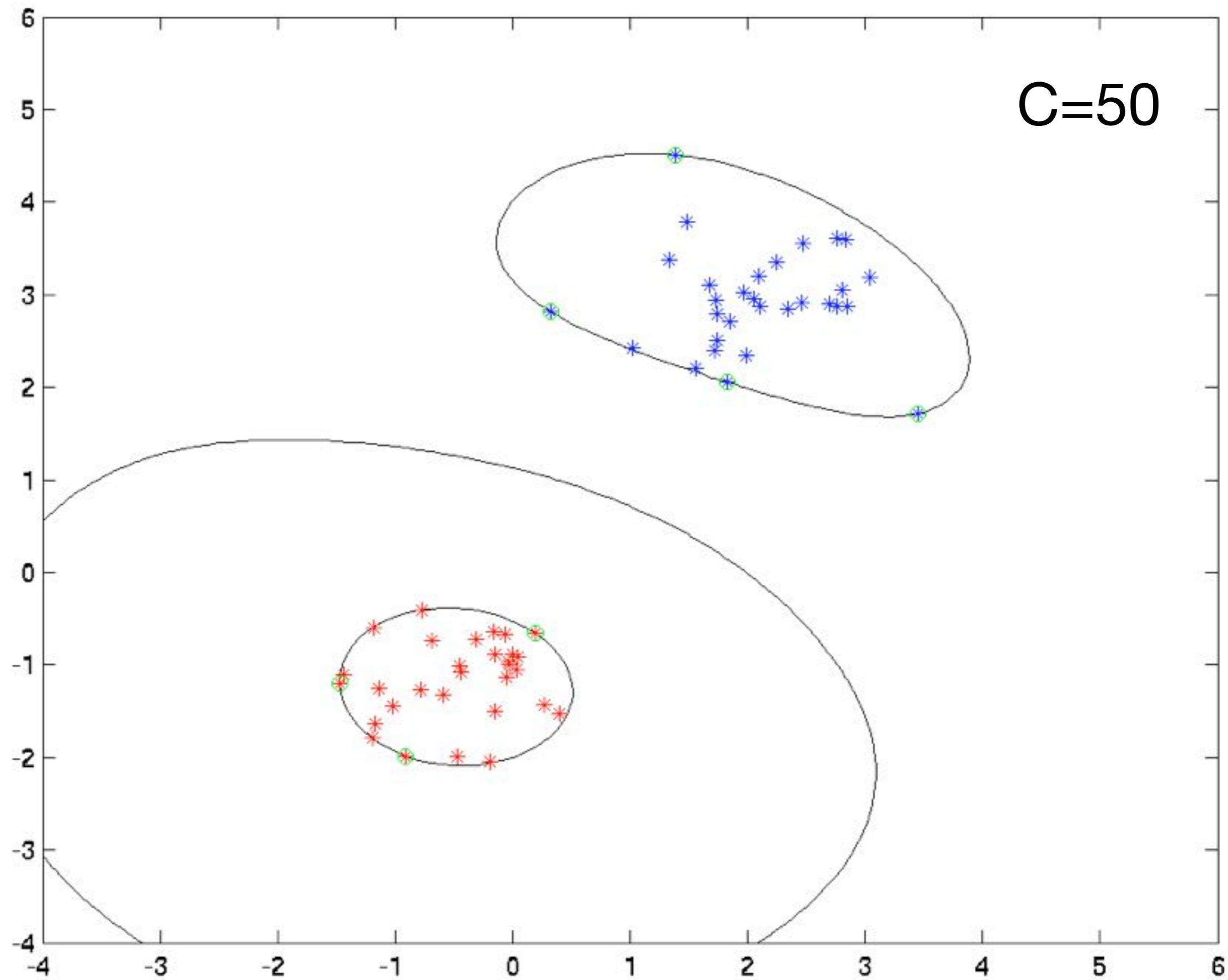
C=10



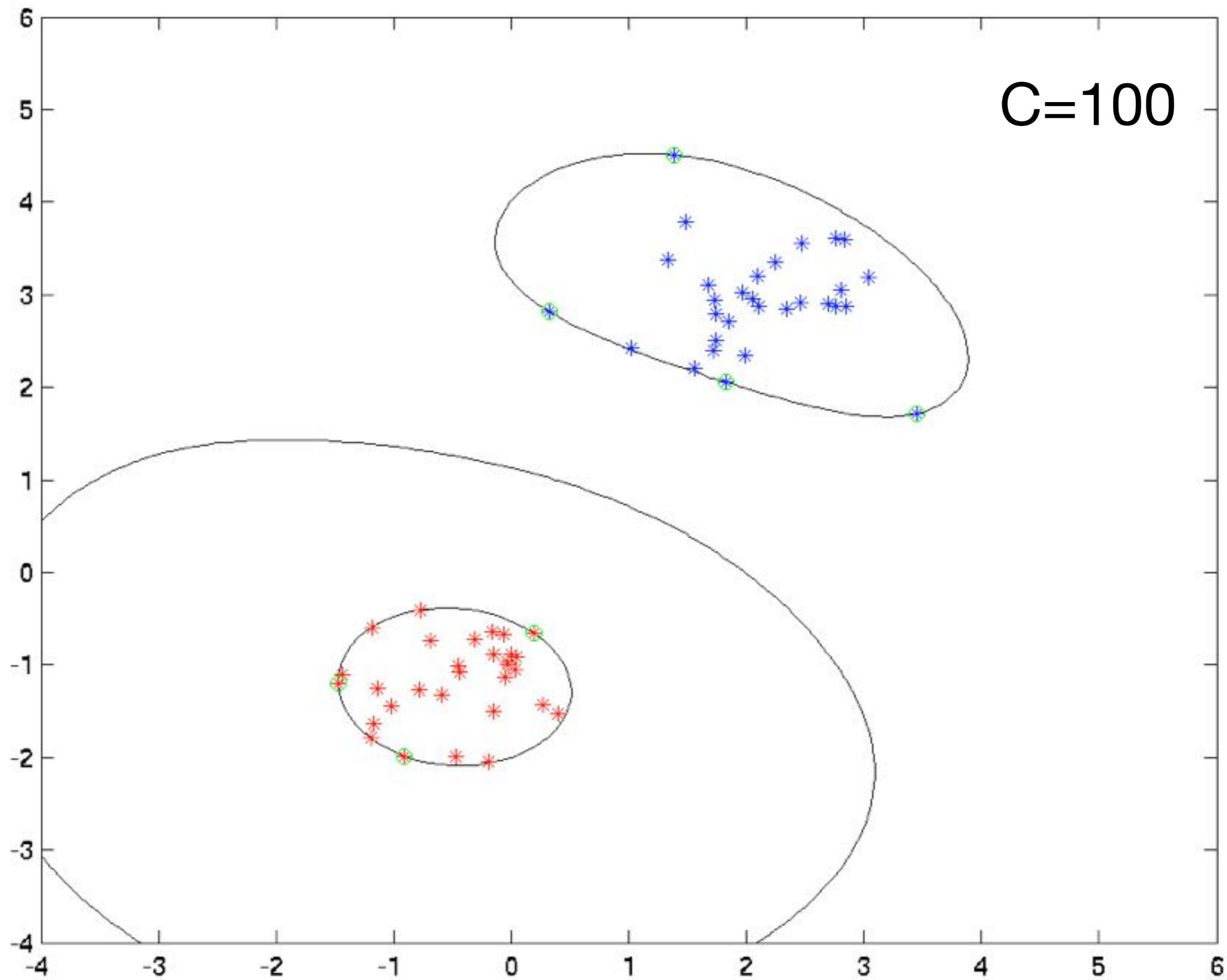
C=20



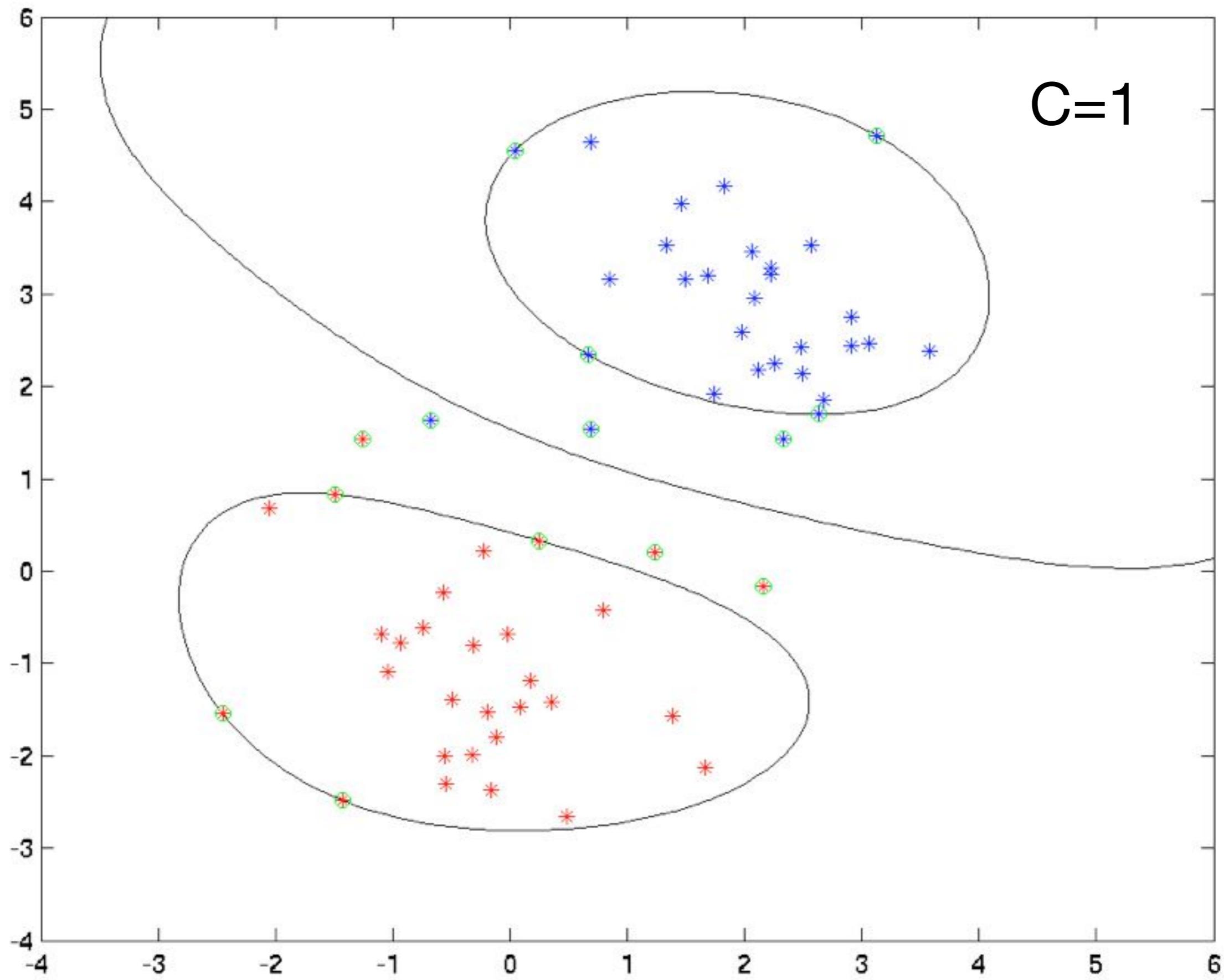
C=50



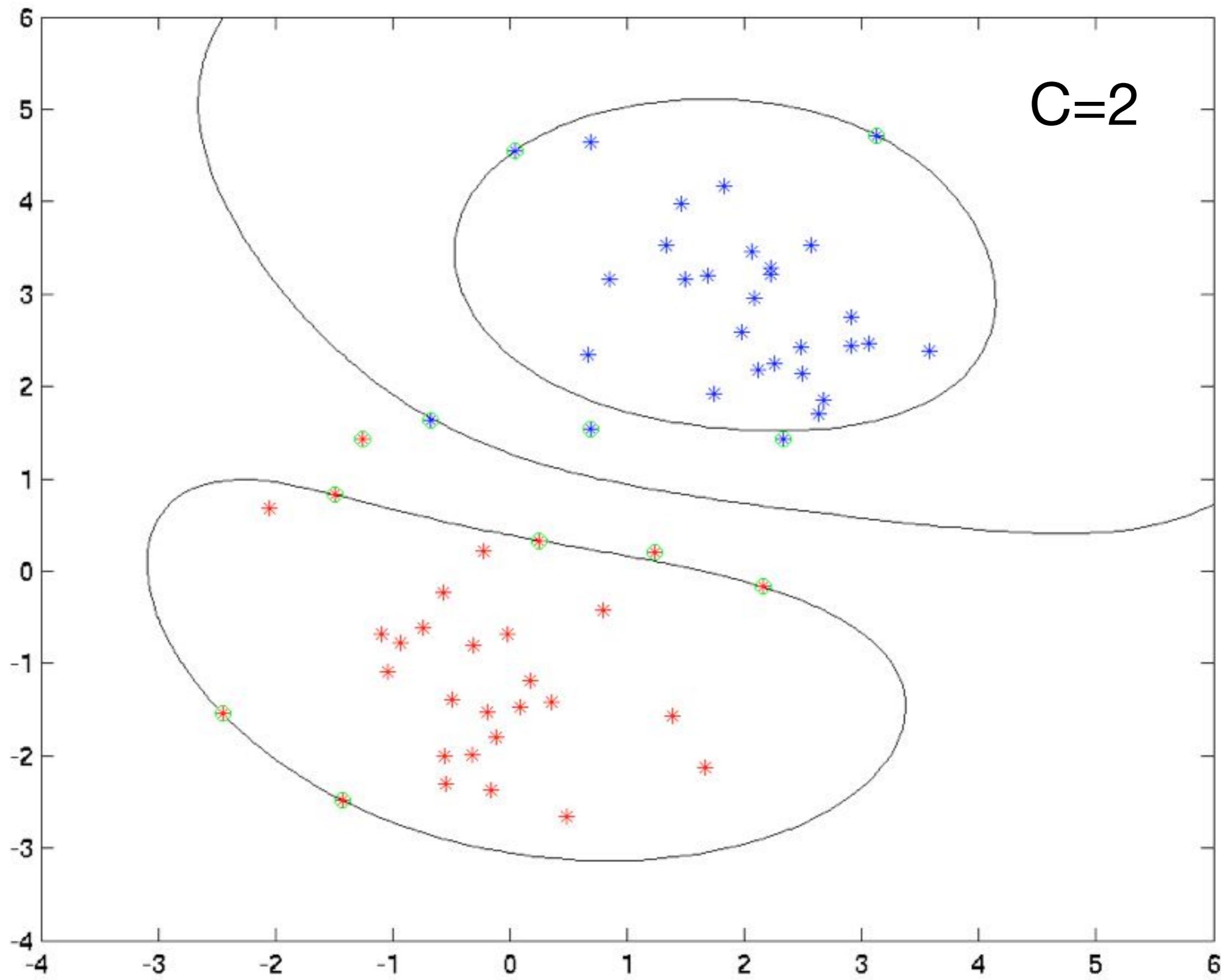
$C=100$



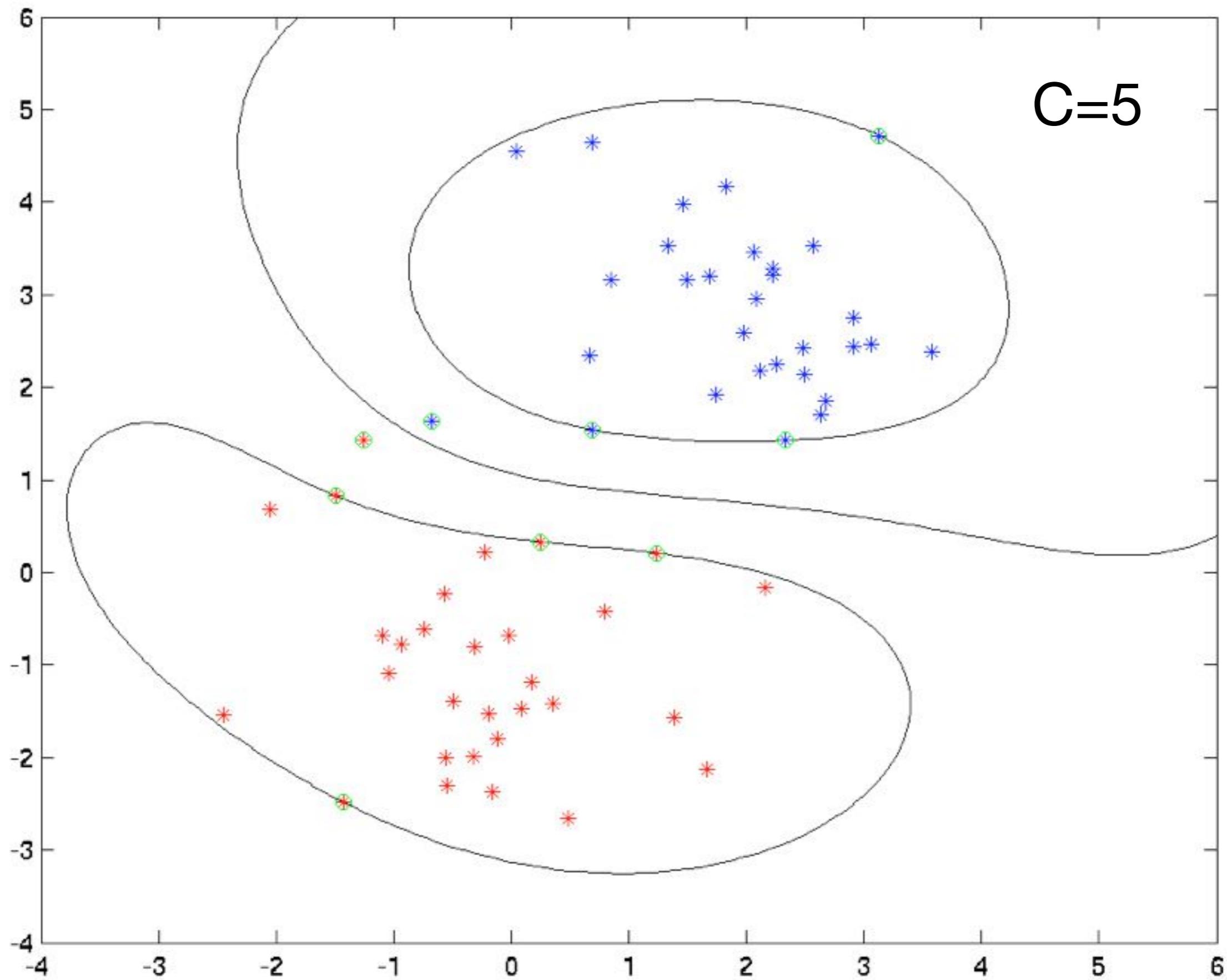
C=1



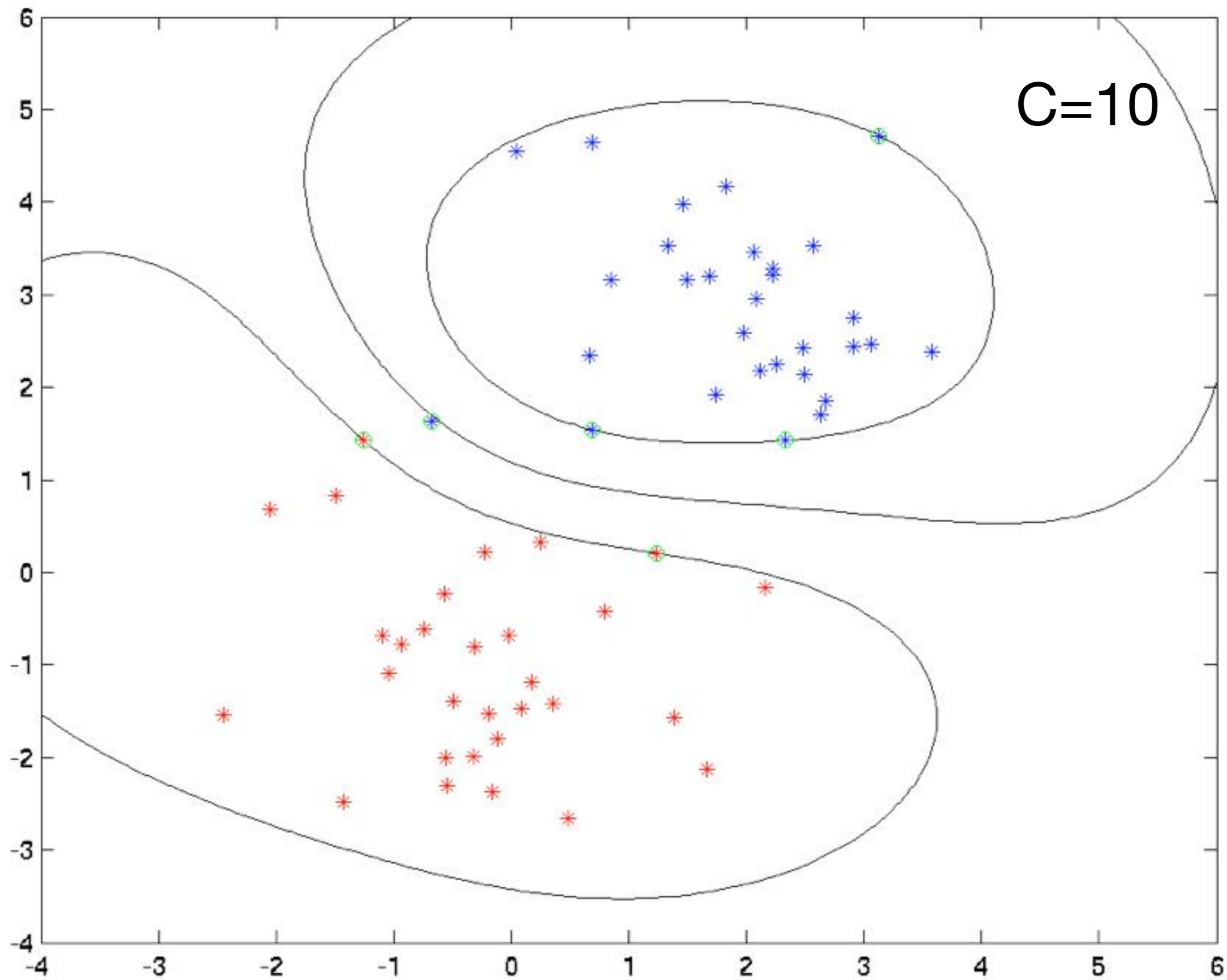
C=2



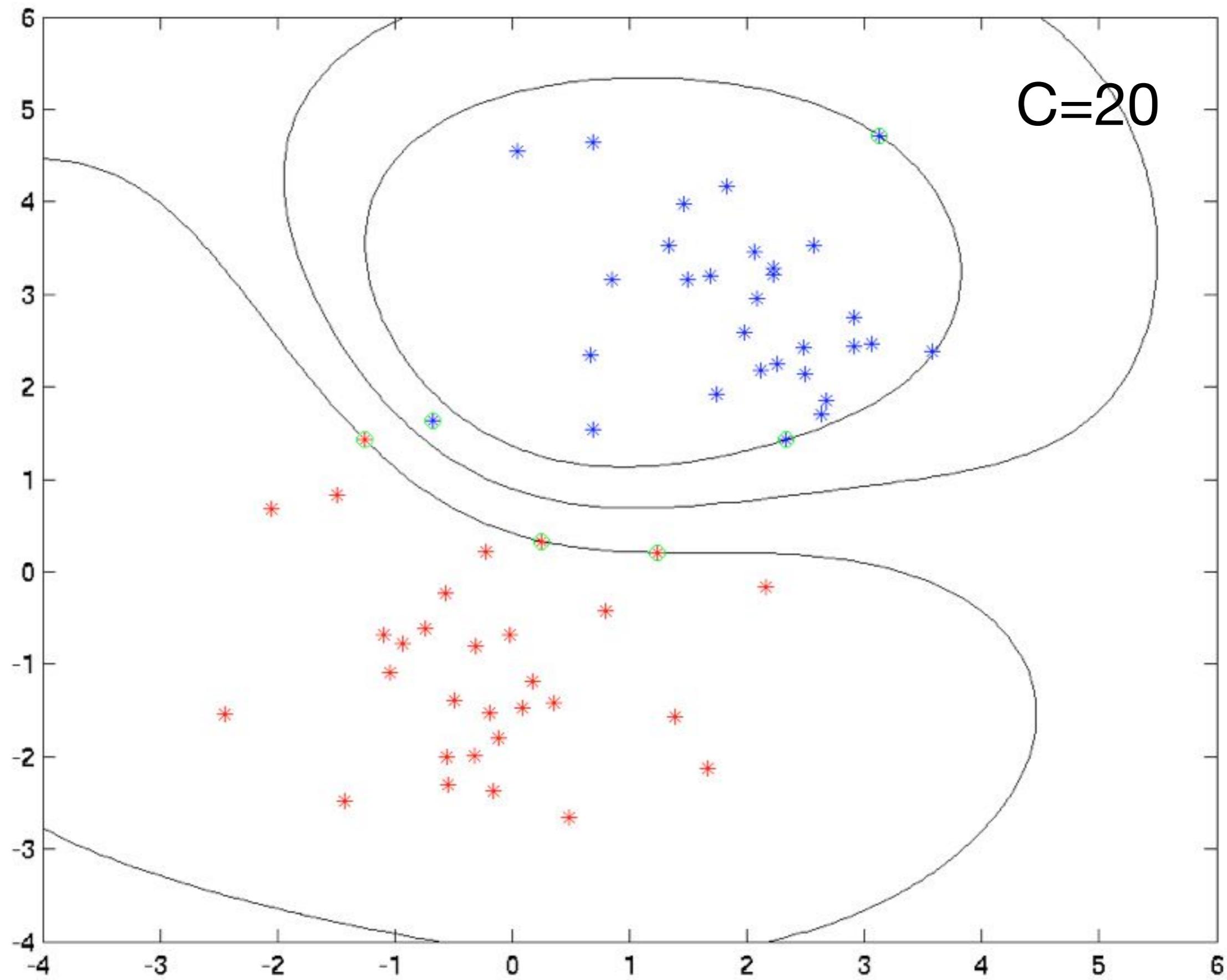
C=5



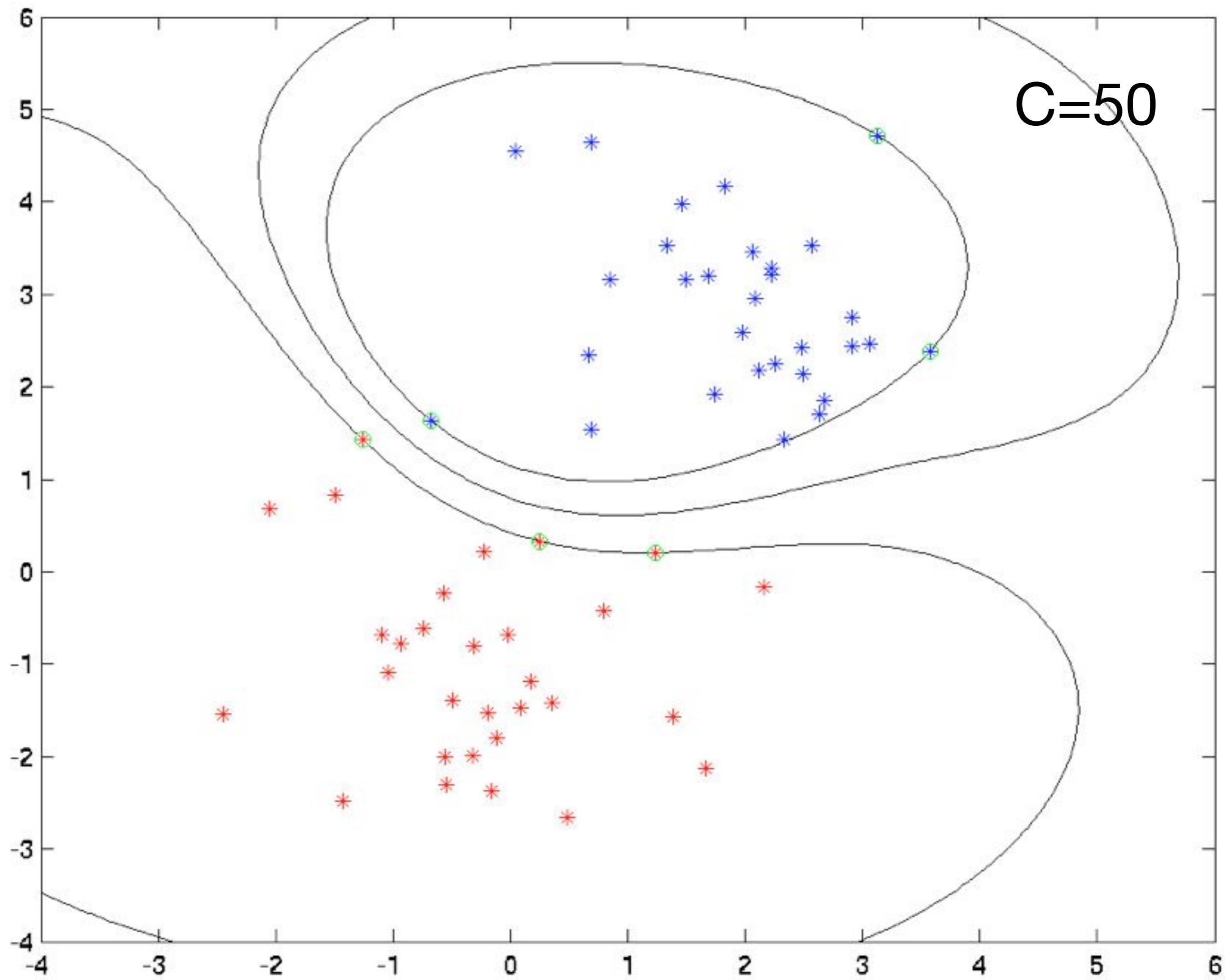
C=10



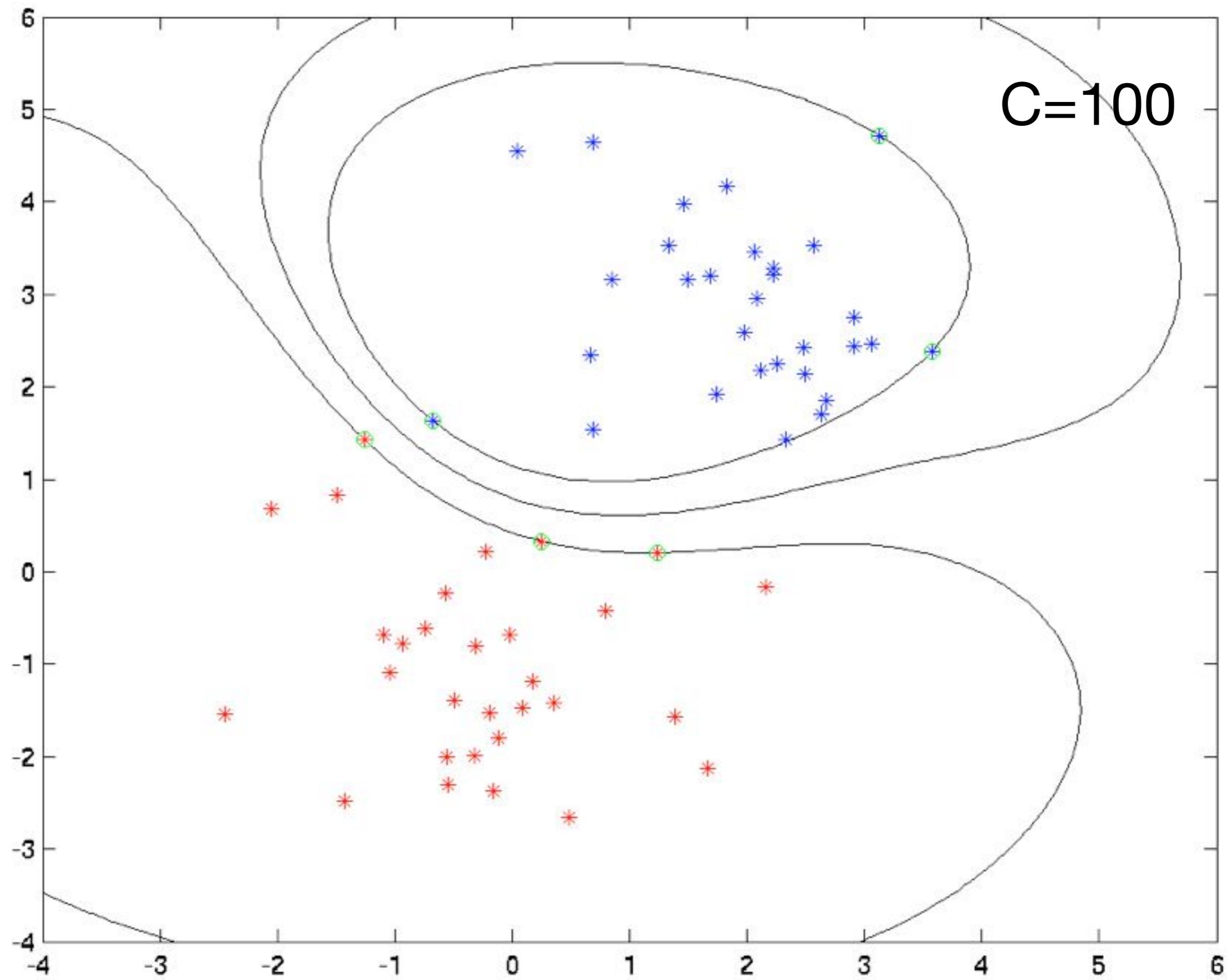
C=20



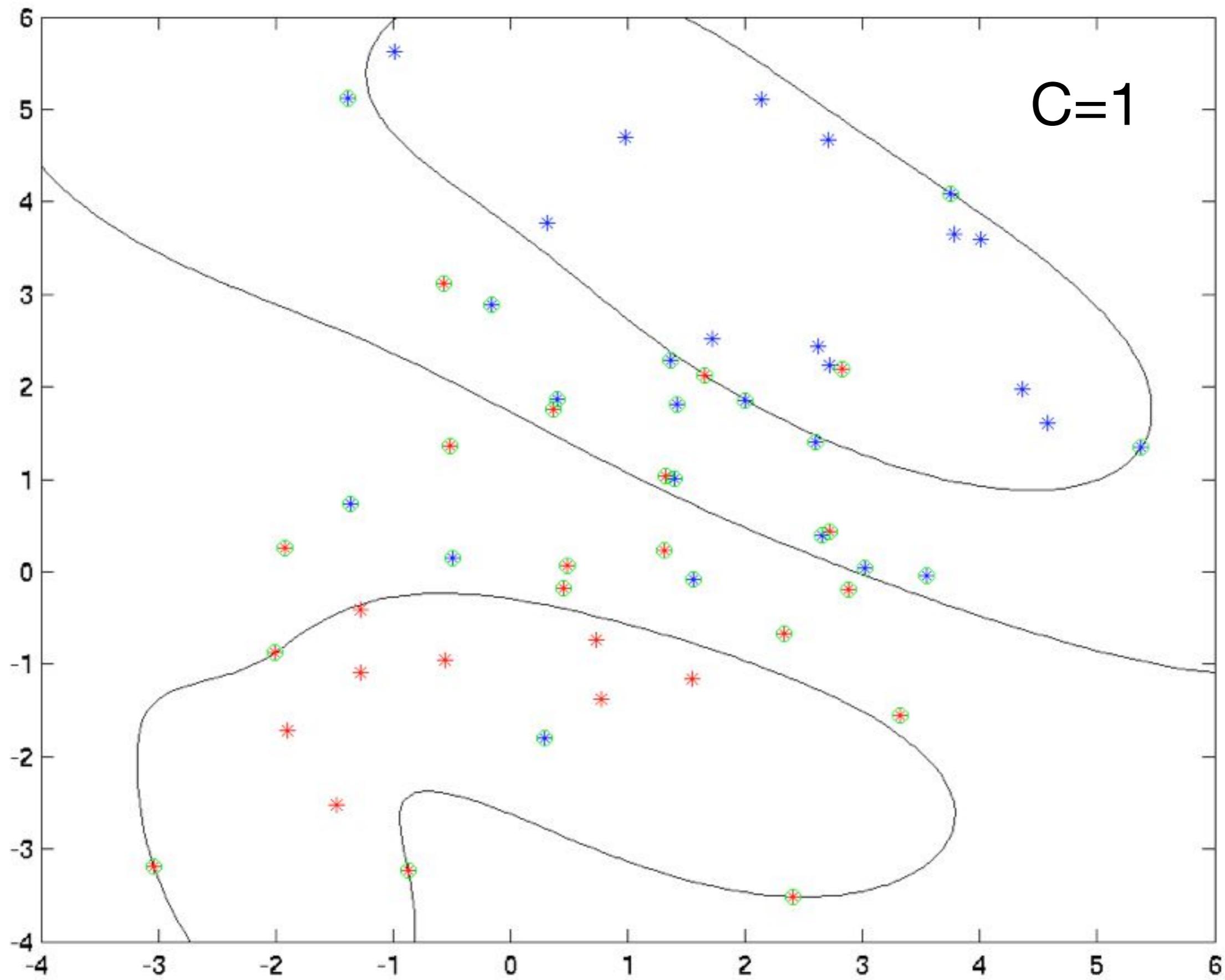
C=50



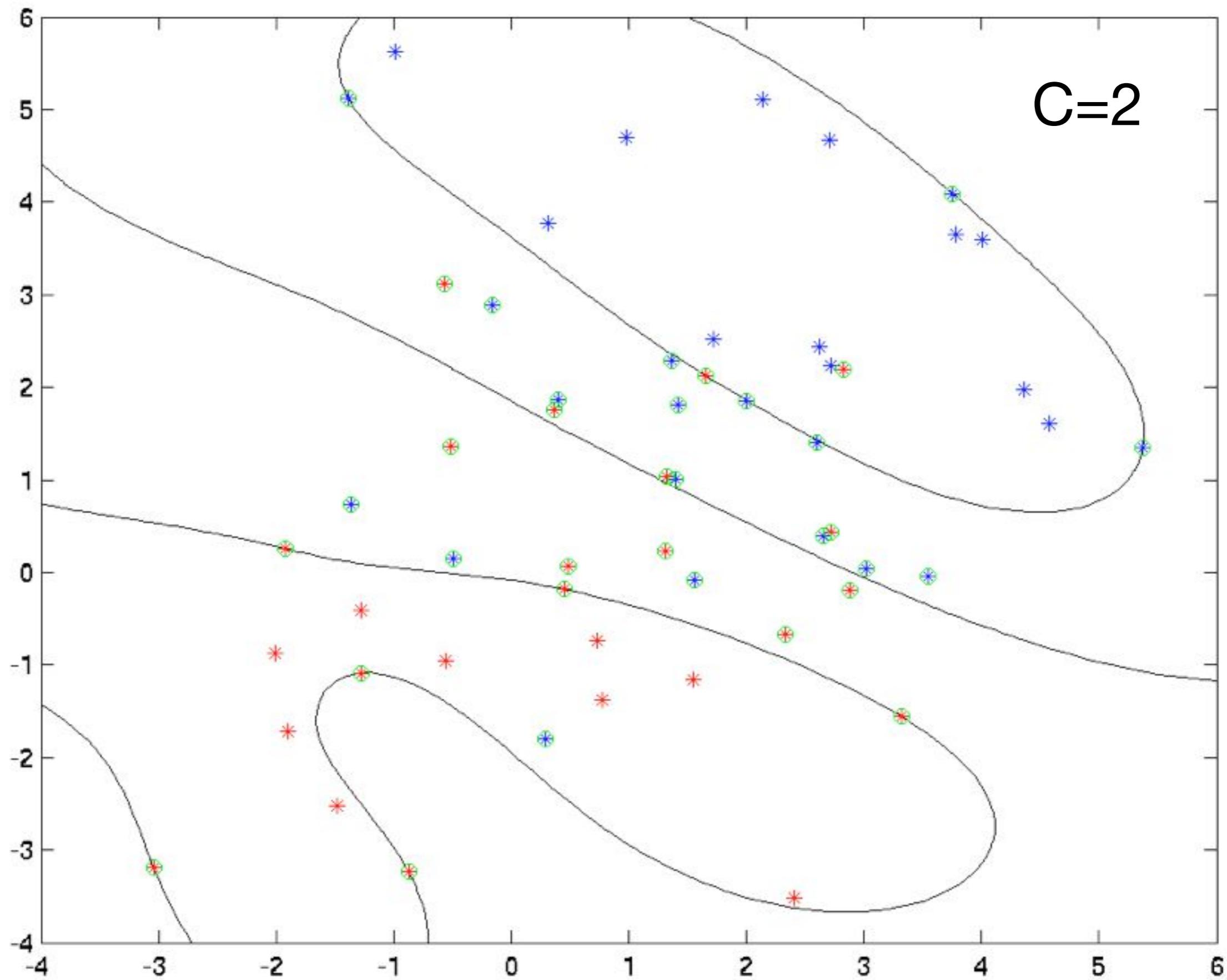
C=100



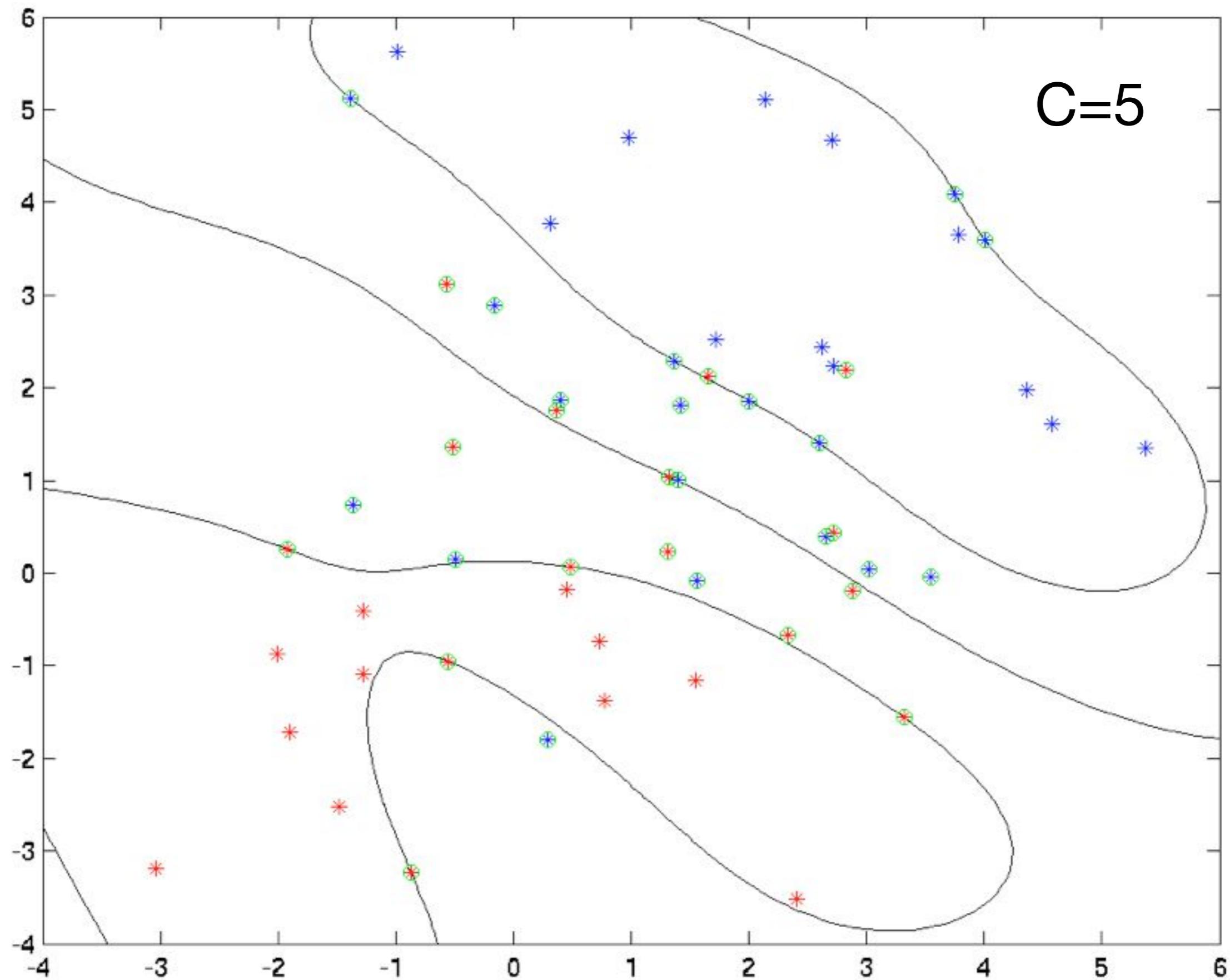
C=1



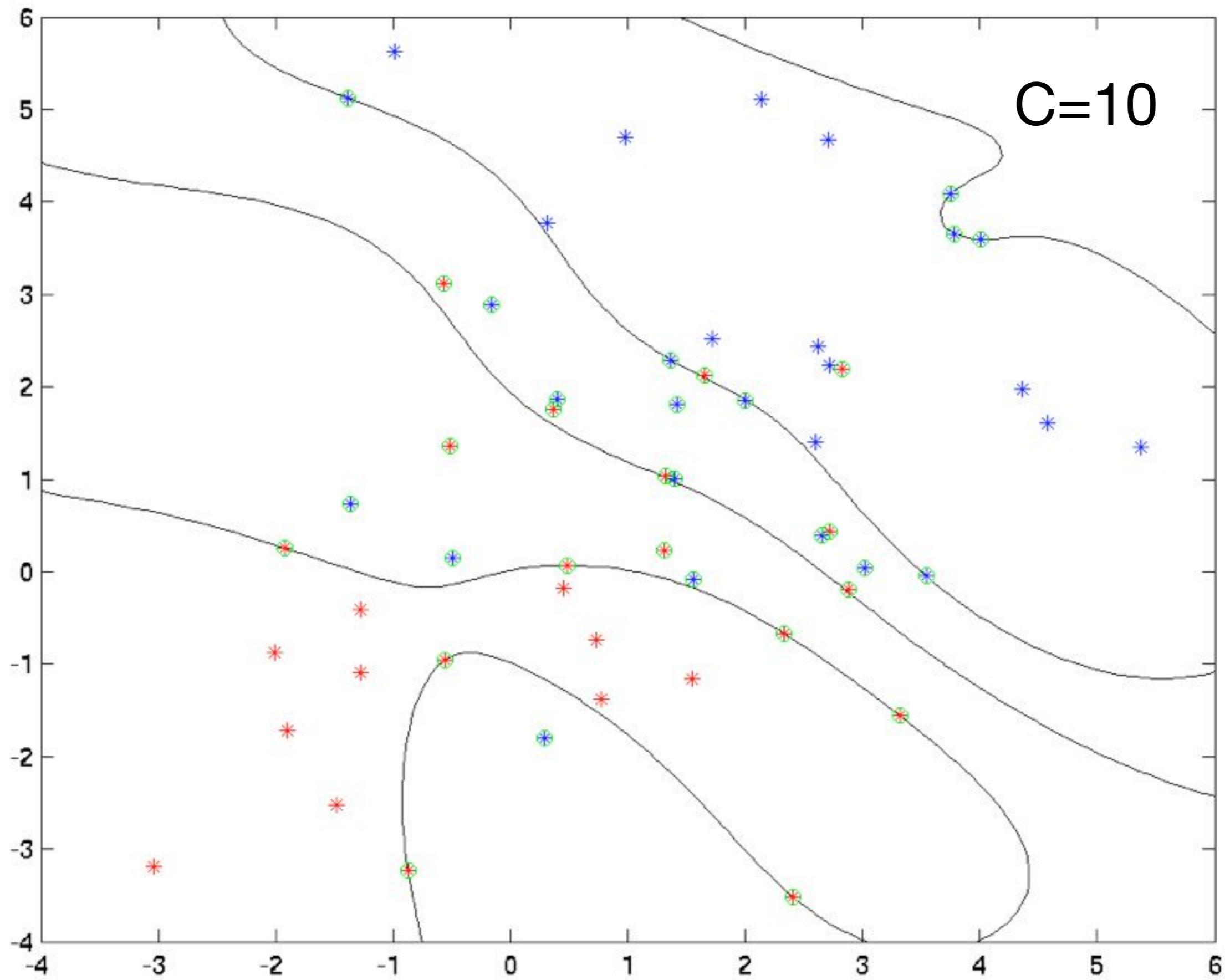
C=2



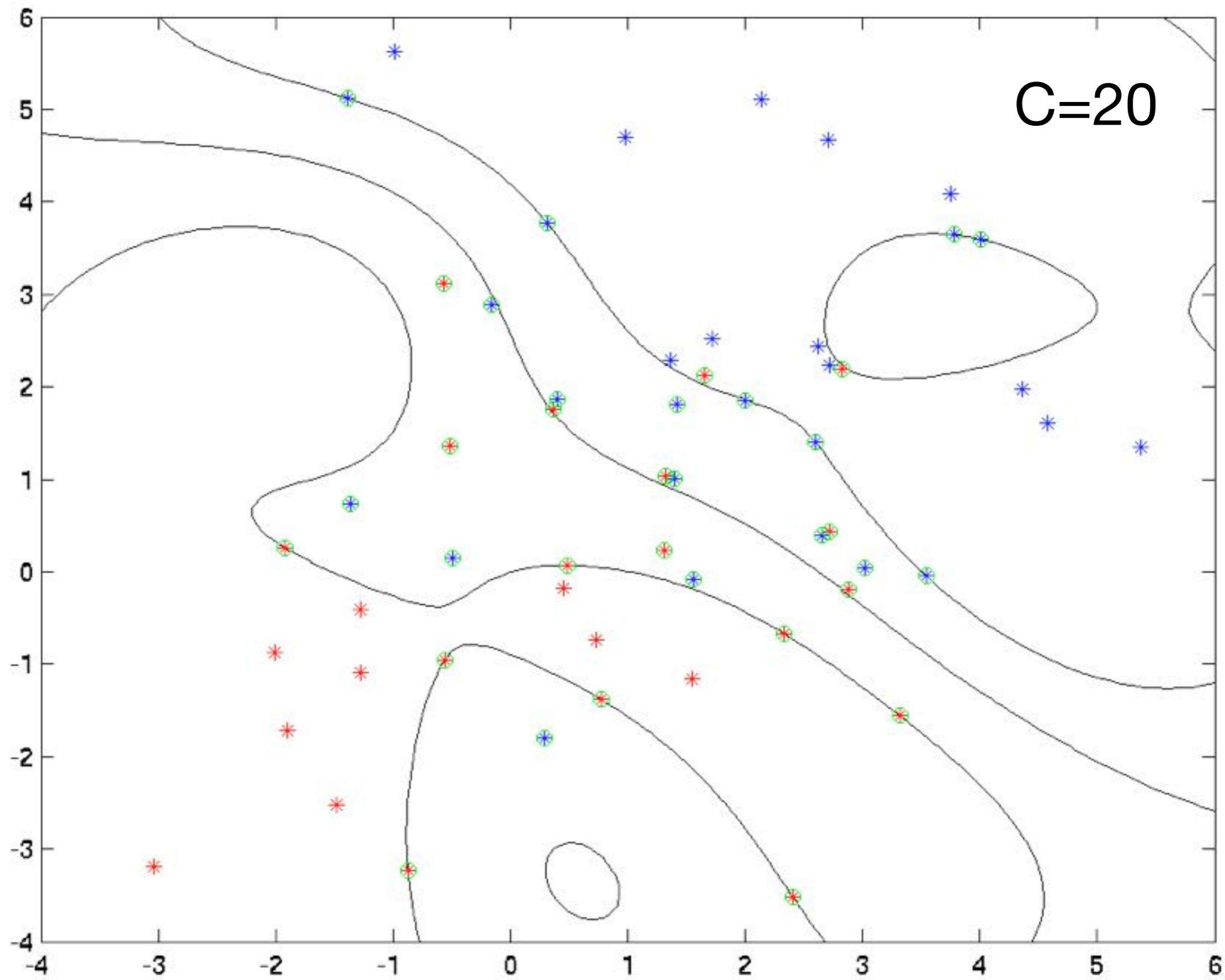
C=5



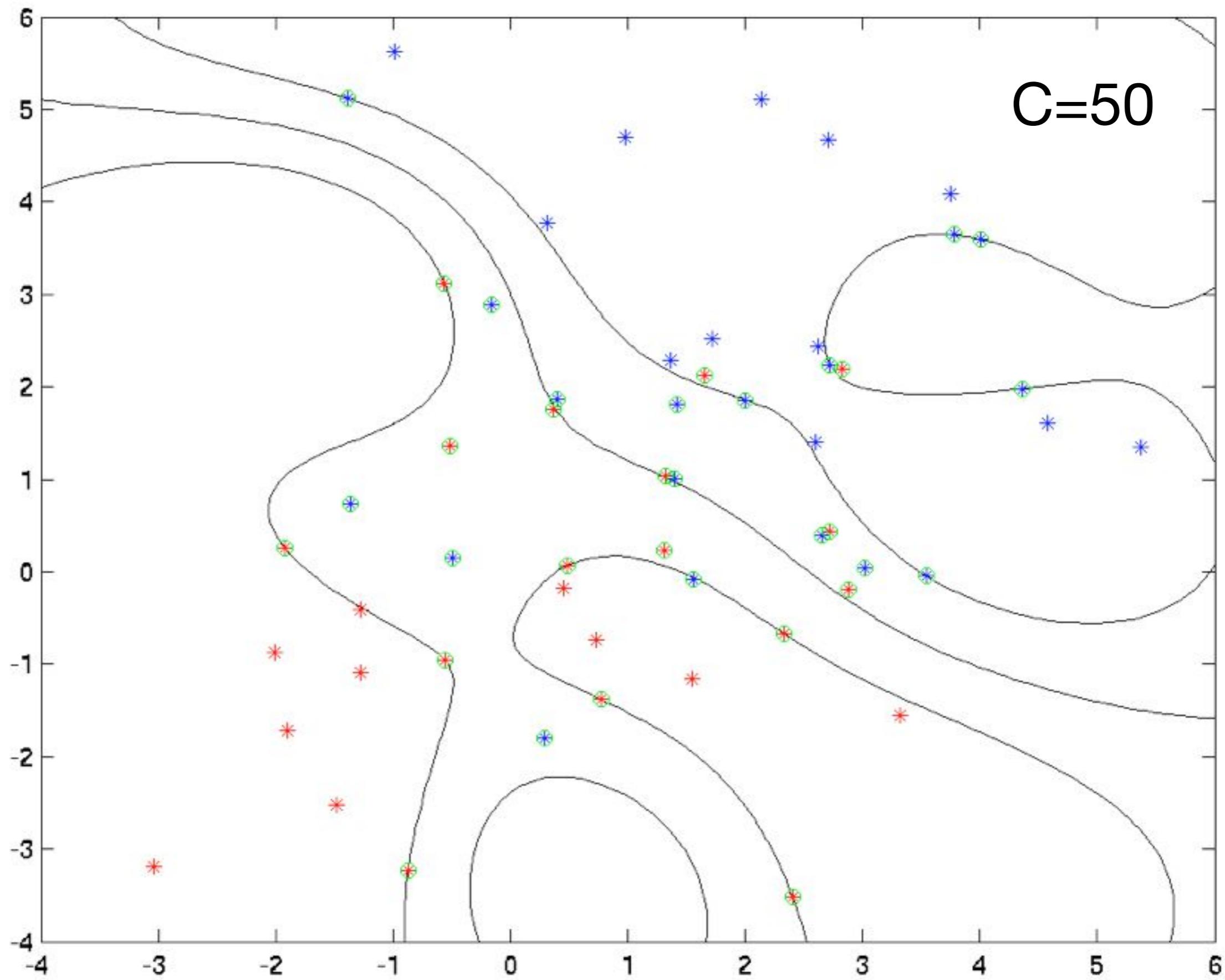
C=10



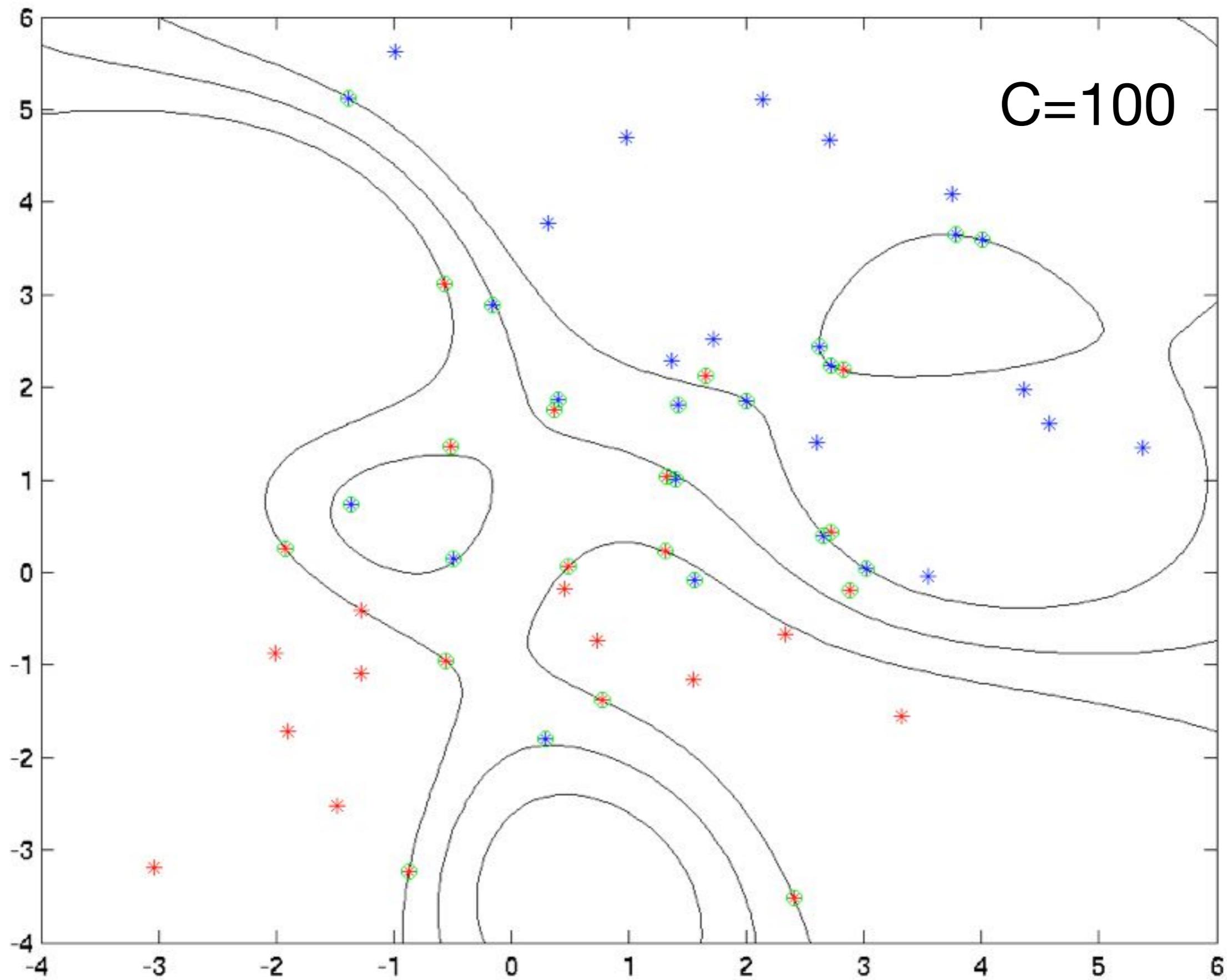
C=20



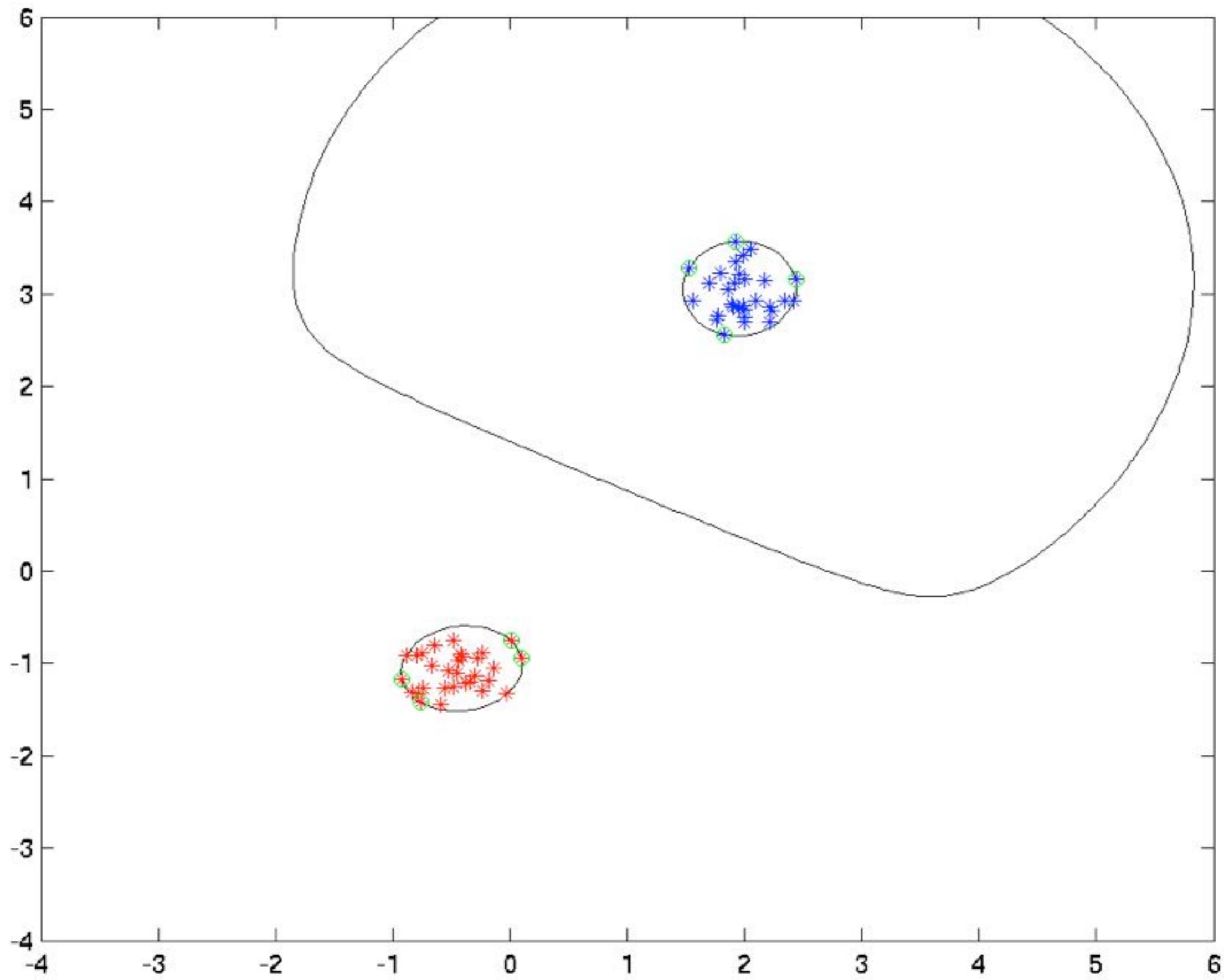
C=50

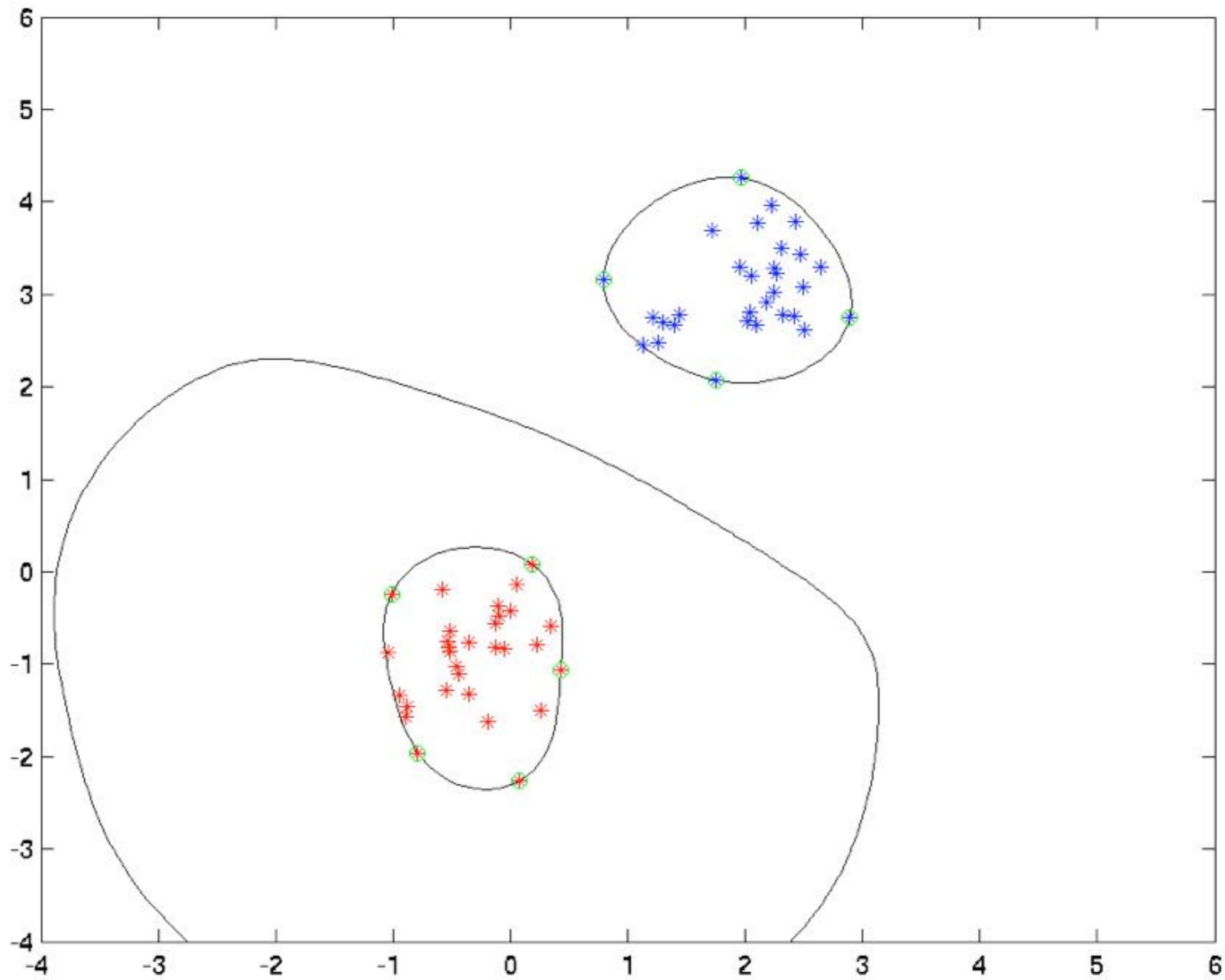


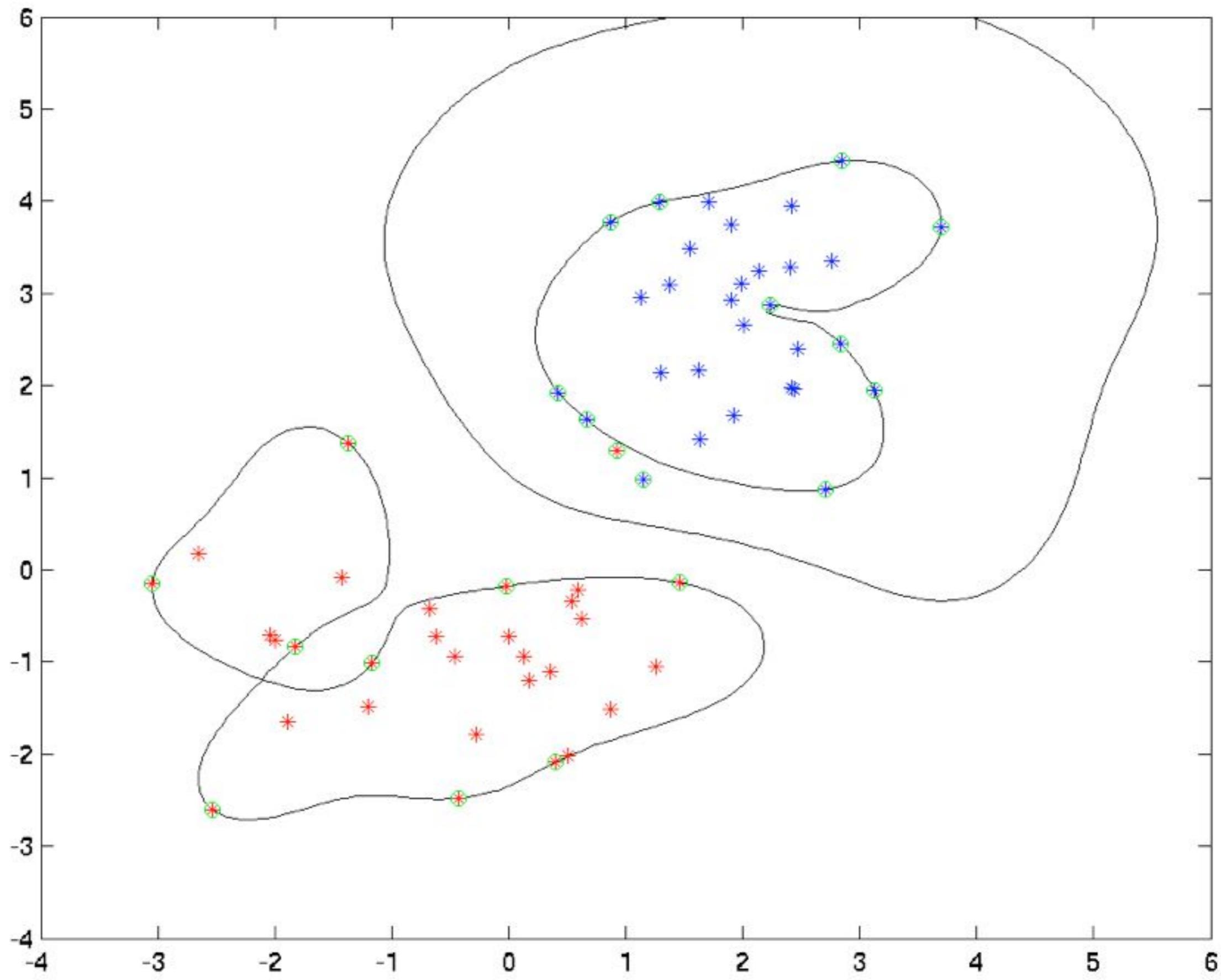
C=100

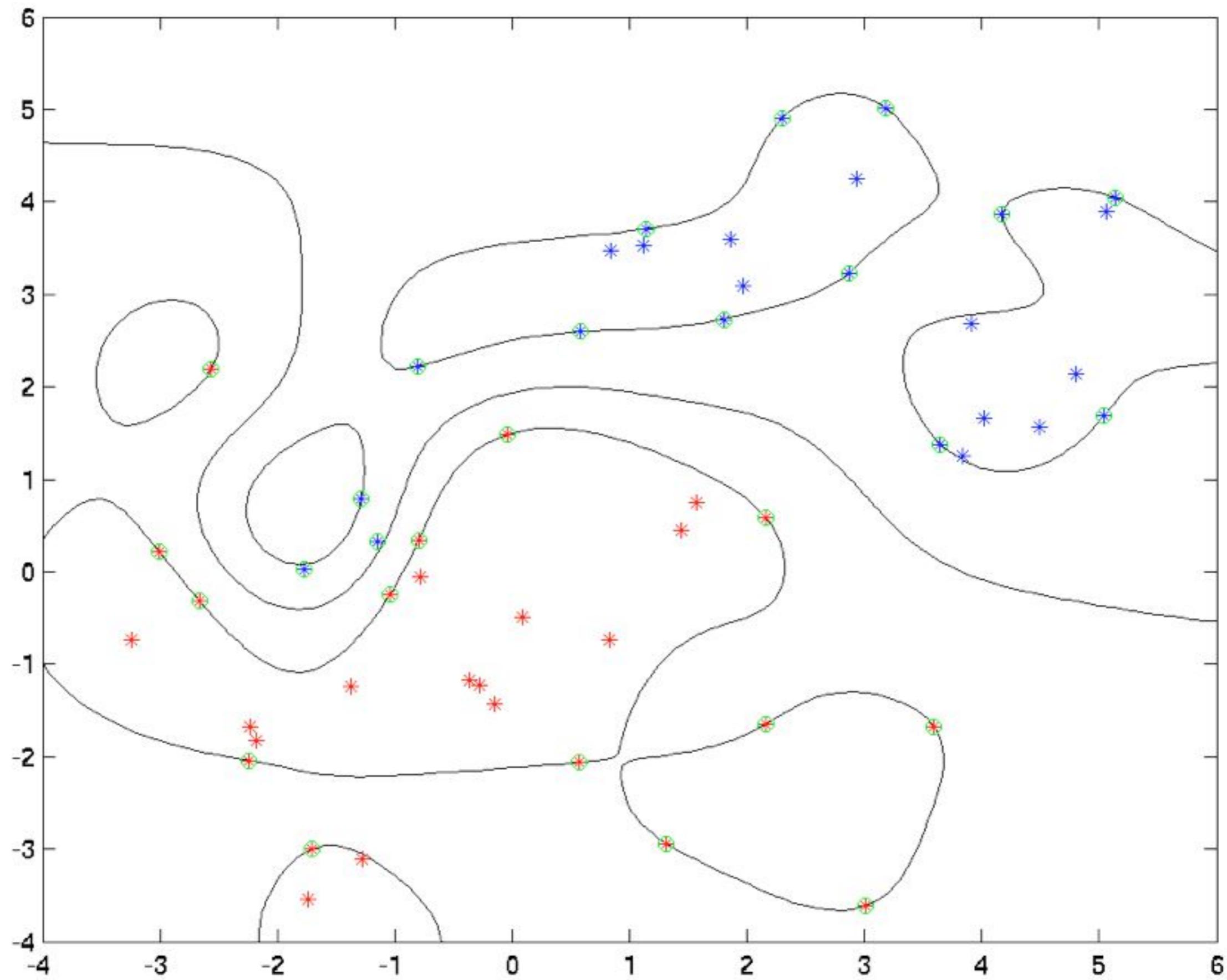


And now with a
narrower kernel

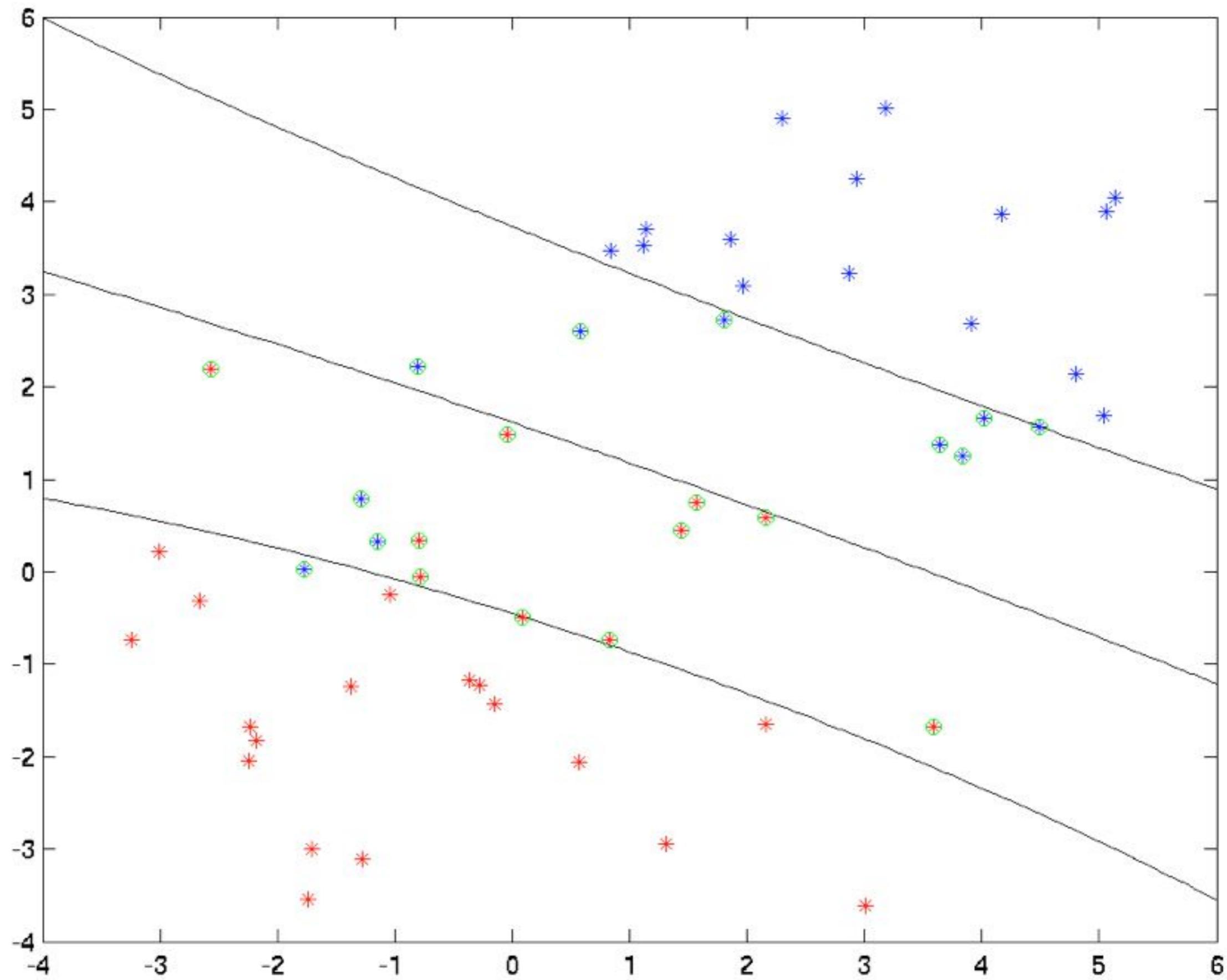




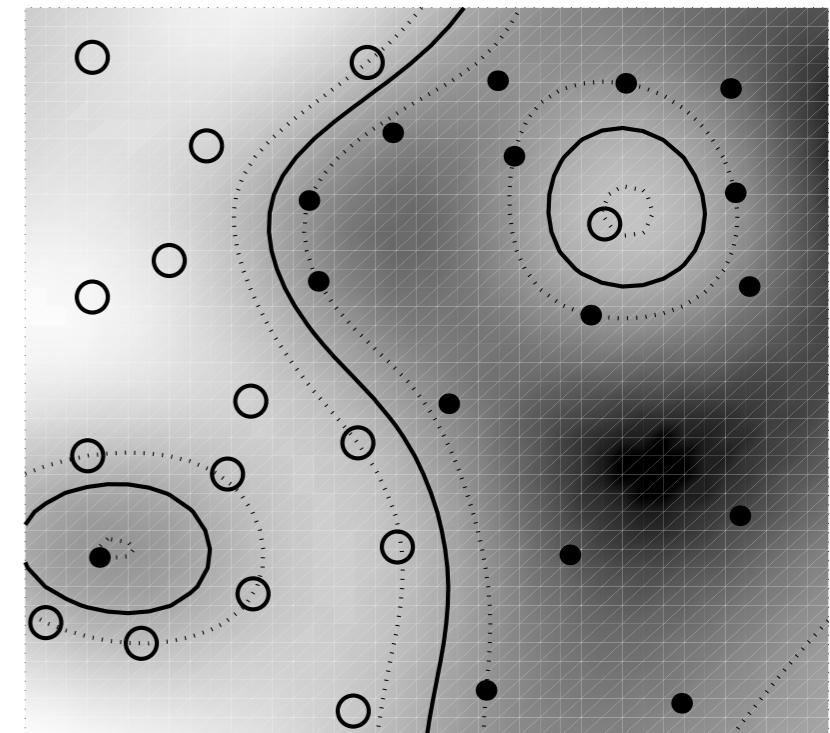
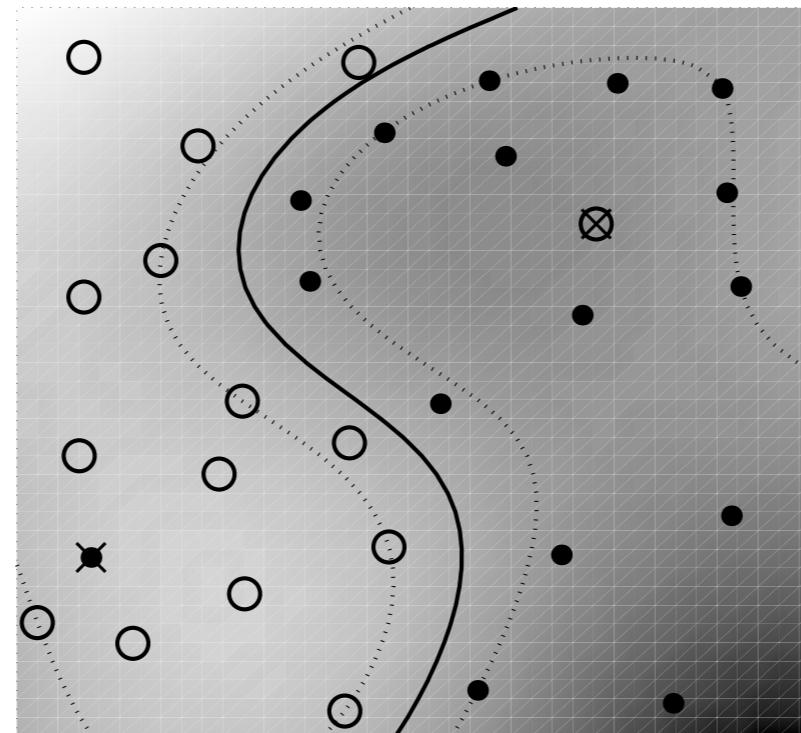
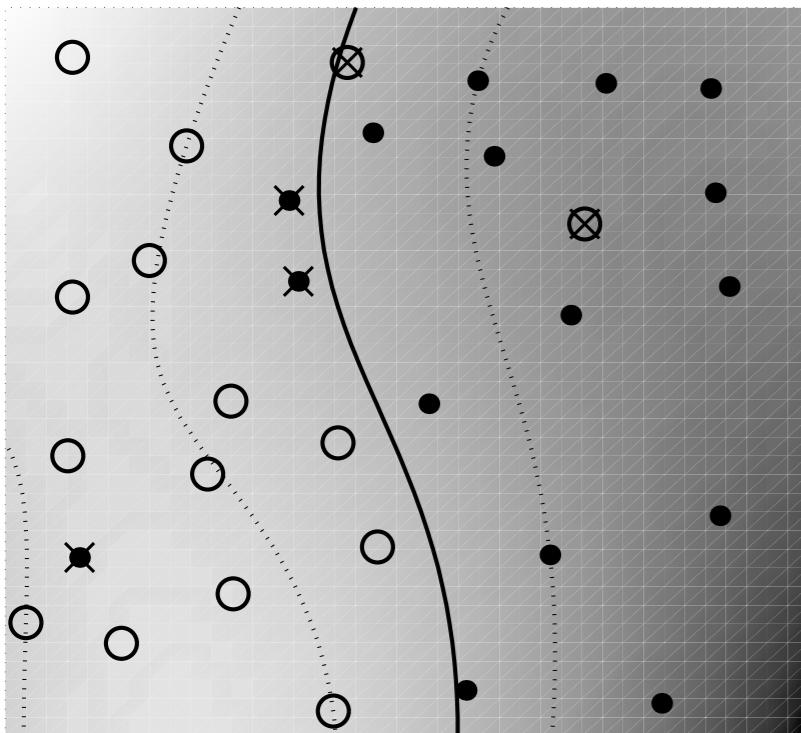




And now with a very
wide kernel



Nonlinear Separation



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

Overfitting?

- Huge feature space with kernels: should we worry about overfitting?
- SVM objective seeks a solution with large margin
 - Theory says that large margin leads to good generalization (we will see this in a couple of lectures)
- But everything overfits sometimes!!!
- Can control by:
 - Setting C
 - Choosing a better Kernel
 - Varying parameters of the Kernel (width of Gaussian, etc.)

Risk and Loss

Loss function point of view

- Constrained quadratic program

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

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

- Risk minimization setting

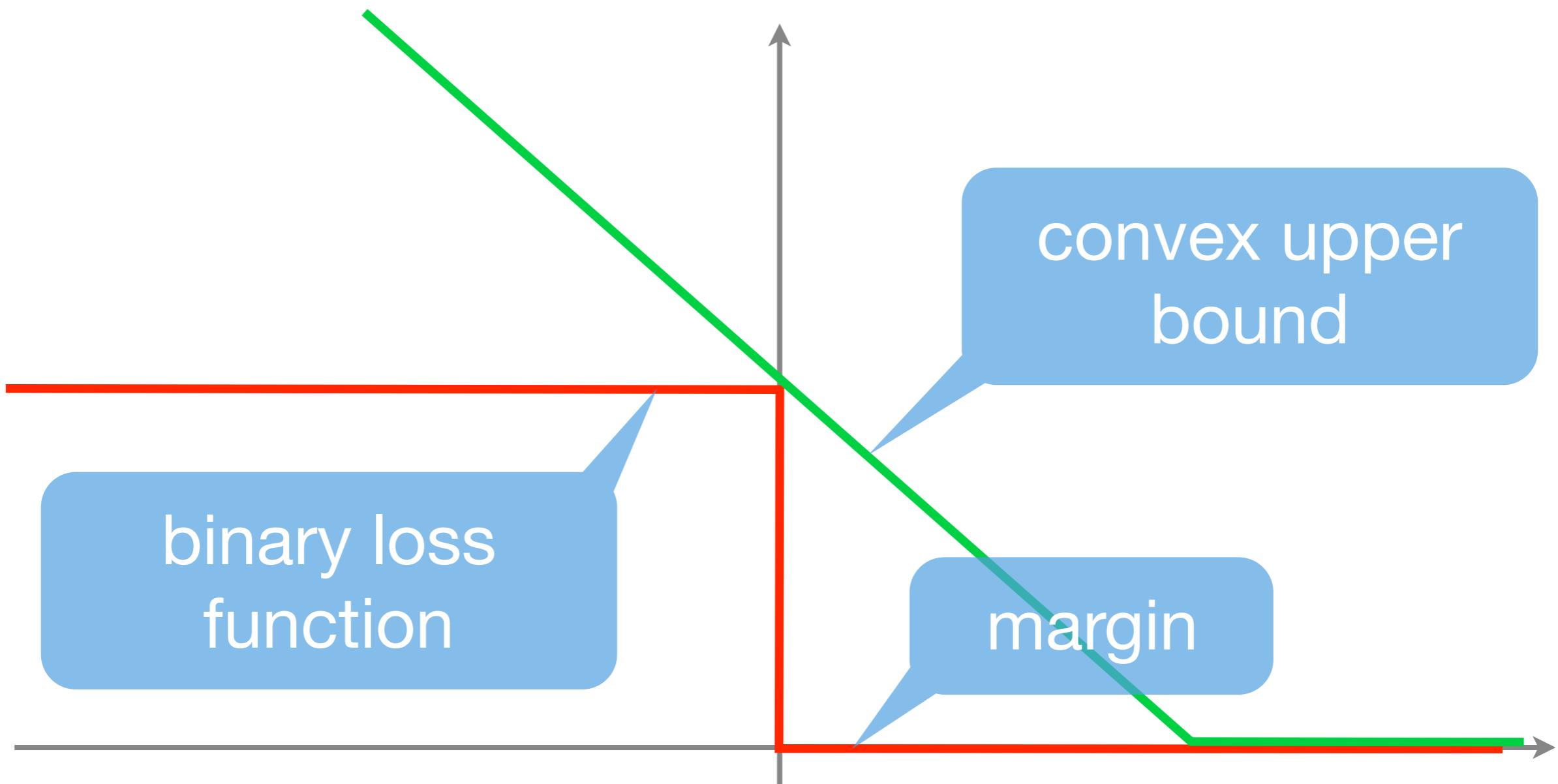
$$\underset{w,b}{\text{minimize}} \quad \frac{1}{2} \|w\|^2 + C \sum_i \max [0, 1 - y_i [\langle w, x_i \rangle + b]]$$

empirical risk

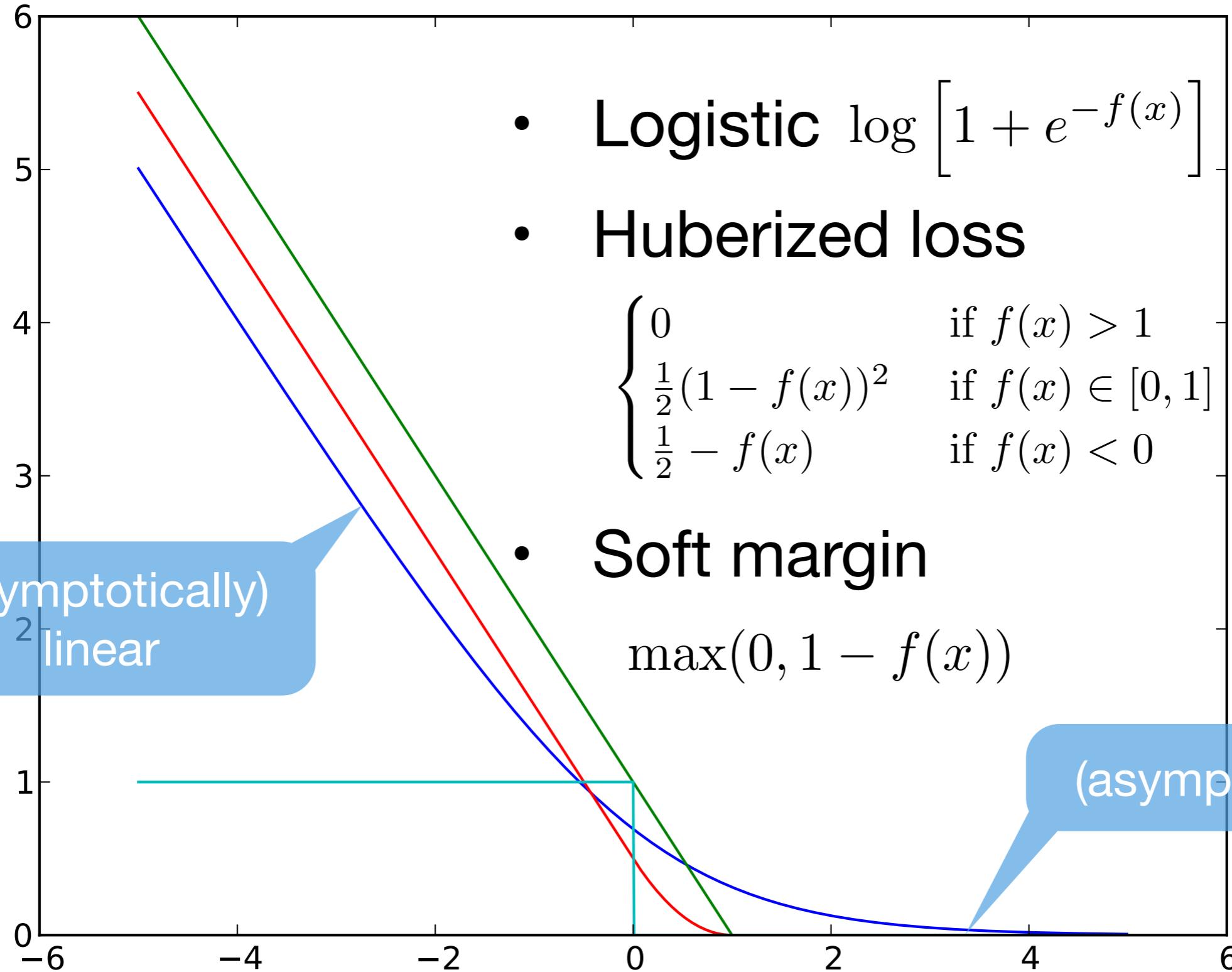
Follows from finding minimal slack variable
for given (w,b) pair.

Soft margin as proxy for binary

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



More loss functions



Risk minimization view

- Find function f minimizing classification error

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

- 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

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

regularization

how to control λ

Support Vector Regression

Regression Estimation

- Find function f minimizing regression error

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

- Compute empirical average

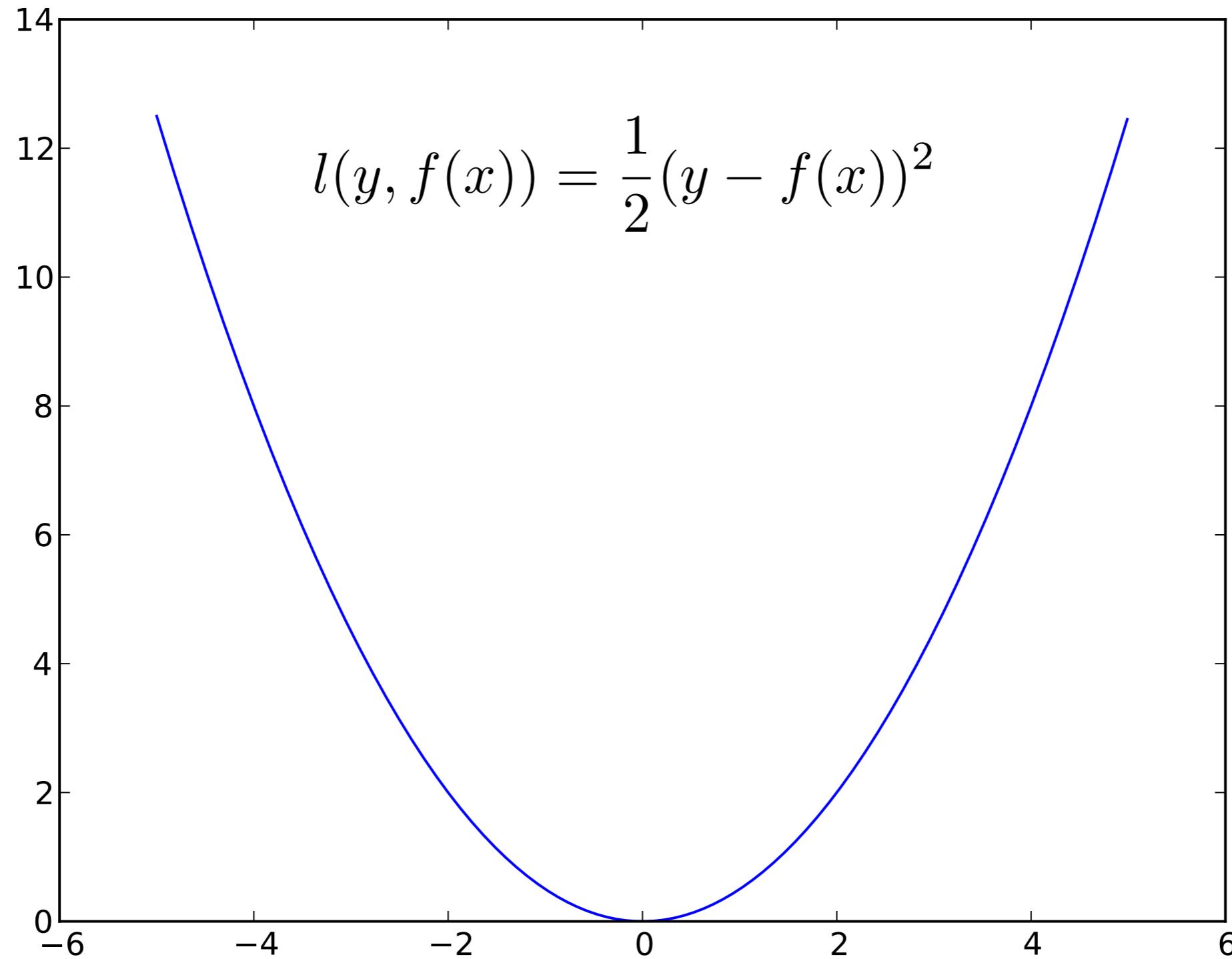
$$R_{\text{emp}}[f] := \frac{1}{m} \sum_{i=1}^m l(y_i, f(x_i))$$

Overfitting as we minimize empirical error

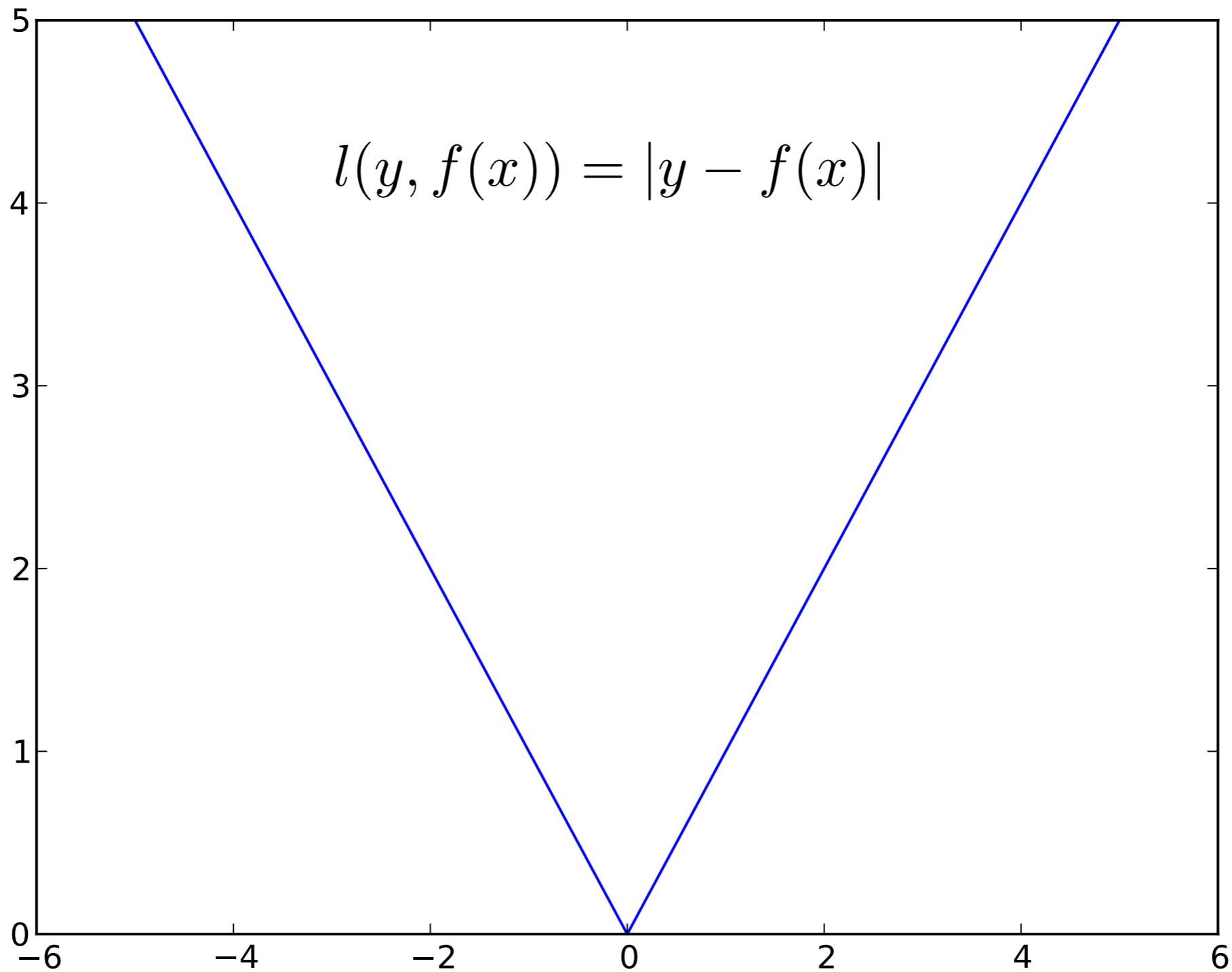
- Add regularization for capacity control

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

Squared loss

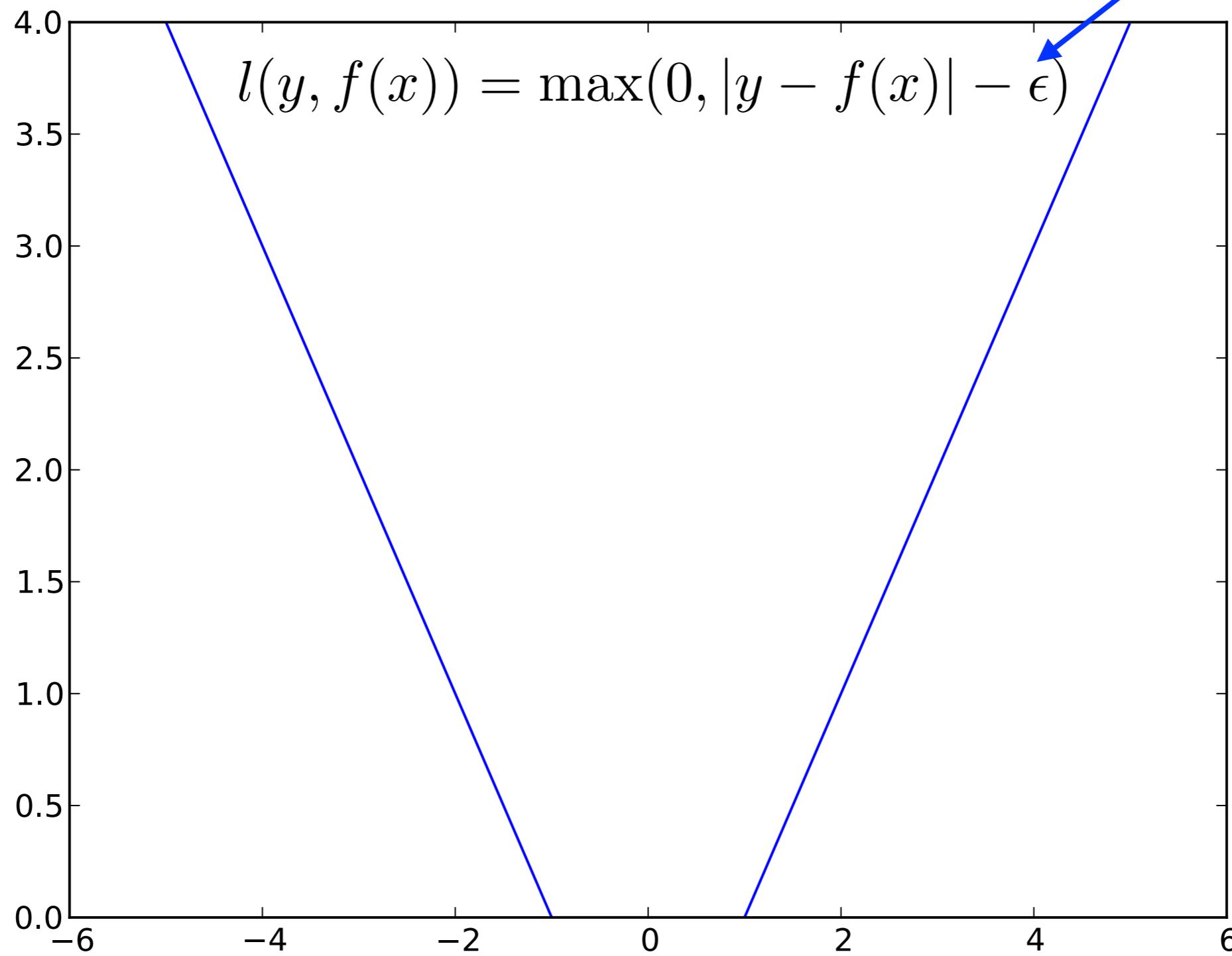


|1 loss



ϵ -insensitive Loss

allow some deviation
without a penalty



Penalized least mean squares

- Optimization problem

$$\underset{w}{\text{minimize}} \frac{1}{2m} \sum_{i=1}^m (y_i - \langle x_i, w \rangle)^2 + \frac{\lambda}{2} \|w\|^2$$

- Solution

$$\begin{aligned}\partial_w [\dots] &= \frac{1}{m} \sum_{i=1}^m [x_i x_i^\top w - x_i y_i] + \lambda w \\ &= \left[\frac{1}{m} X X^\top + \lambda \mathbf{1} \right] w - \frac{1}{m} X y = 0\end{aligned}$$

$$\text{hence } w = [X X^\top + \lambda m \mathbf{1}]^{-1} X y$$

Outer product
matrix in X

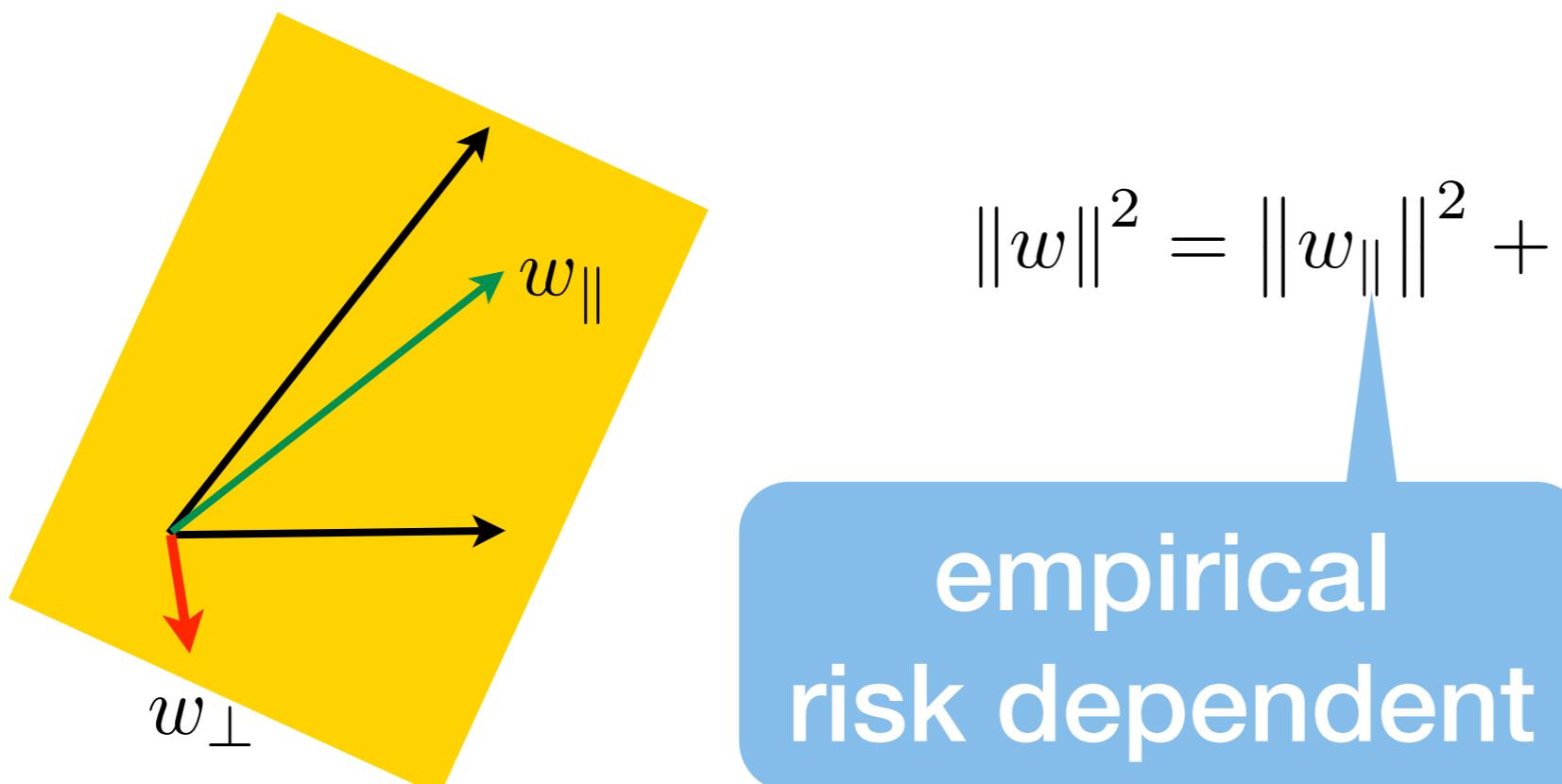
Conjugate Gradient
Sherman Morrison Woodbury

Penalized least mean squares ... now with kernels

- Optimization problem

$$\underset{w}{\text{minimize}} \frac{1}{2m} \sum_{i=1}^m (y_i - \langle \phi(x_i), w \rangle)^2 + \frac{\lambda}{2} \|w\|^2$$

- Representer Theorem (Kimeldorf & Wahba, 1971)



$$\|w\|^2 = \|w_{\parallel}\|^2 + \|w_{\perp}\|^2$$

empirical
risk dependent

Penalized least mean squares ... now with kernels

- Optimization problem

$$\underset{w}{\text{minimize}} \frac{1}{2m} \sum_{i=1}^m (y_i - \langle \phi(x_i), w \rangle)^2 + \frac{\lambda}{2} \|w\|^2$$

- Representer Theorem (Kimeldorf & Wahba, 1971)
 - Optimal solution is in span of data $w = \sum \alpha_i \phi(x_i)$
 - Proof - risk term only depends on data via $\phi(x_i)$
 - Regularization ensures that orthogonal part is 0
- Optimization problem in terms of w

$$\underset{\alpha}{\text{minimize}} \frac{1}{2m} \sum_{i=1}^m \left(y_i - \sum_j K_{ij} \alpha_j \right)^2 + \frac{\lambda}{2} \sum_{i,j} \alpha_i \alpha_j K_{ij}$$

solve for $\alpha = (K + m\lambda \mathbf{1})^{-1} y$ as linear system

Penalized least mean squares ... now with kernels

- Optimization problem

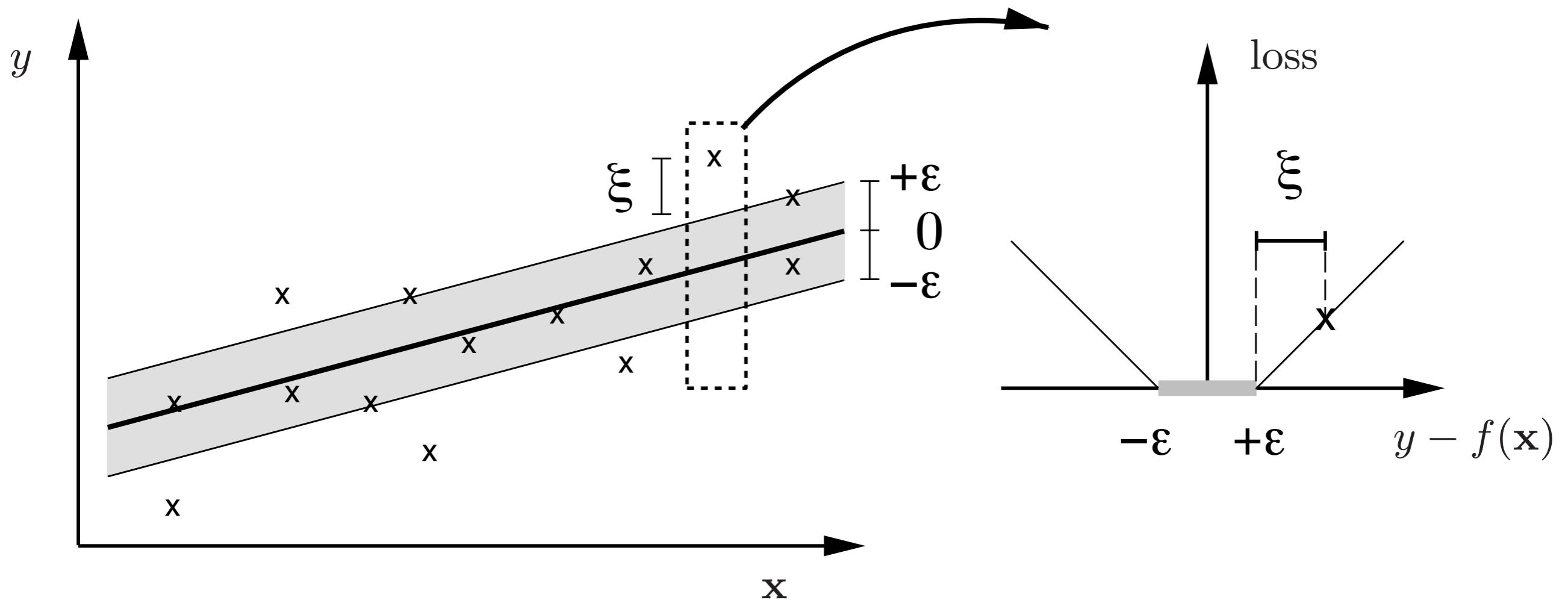
$$\underset{w}{\text{minimize}} \frac{1}{2m} \sum_{i=1}^m (y_i - \langle \phi(x_i), w \rangle)^2 + \frac{\lambda}{2} \|w\|^2$$

- Representer Theorem (Kimeldorf & Wahba, 1971)
 - Optimal solution is in span of data $w = \sum_i \alpha_i \phi(x_i)$
 - Proof - risk term only depends on data via $\phi(x_i)$
 - Regularization ensures that orthogonal part is 0
- Optimization problem in terms of w

$$\underset{\alpha}{\text{minimize}} \frac{1}{2m} \sum_{i=1}^m \left(y_i - \sum_j K_{ij} \alpha_j \right)^2 + \frac{\lambda}{2} \sum_{i,j} \alpha_i \alpha_j K_{ij}$$

solve for $\alpha = (K + m\lambda I)^{-1} y$ as linear system

SVM Regression (ϵ -insensitive loss)



don't care about deviations within the tube

SVM Regression (ϵ -insensitive loss)

- Optimization Problem (as constrained QP)

$$\underset{w,b}{\text{minimize}} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m [\xi_i + \xi_i^*]$$

subject to $\langle w, x_i \rangle + b \leq y_i + \epsilon + \xi_i$ and $\xi_i \geq 0$

$\langle w, x_i \rangle + b \geq y_i - \epsilon - \xi_i^*$ and $\xi_i^* \geq 0$

- Lagrange Function

$$L = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m [\xi_i + \xi_i^*] - \sum_{i=1}^m [\eta_i \xi_i + \eta_i^* \xi_i^*] +$$

$$\sum_{i=1}^m \alpha_i [\langle w, x_i \rangle + b - y_i - \epsilon - \xi_i] + \sum_{i=1}^m \alpha_i^* [y_i - \epsilon - \xi_i^* - \langle w, x_i \rangle - b]$$

SVM Regression (ϵ -insensitive loss)

- First order conditions

$$\partial_w L = 0 = w + \sum_i [\alpha_i - \alpha_i^*] x_i$$

$$\partial_b L = 0 = \sum_i [\alpha_i - \alpha_i^*]$$

$$\partial_{\xi_i} L = 0 = C - \eta_i - \alpha_i$$

$$\partial_{\xi_i^*} L = 0 = C - \eta_i^* - \alpha_i^*$$

- Dual problem

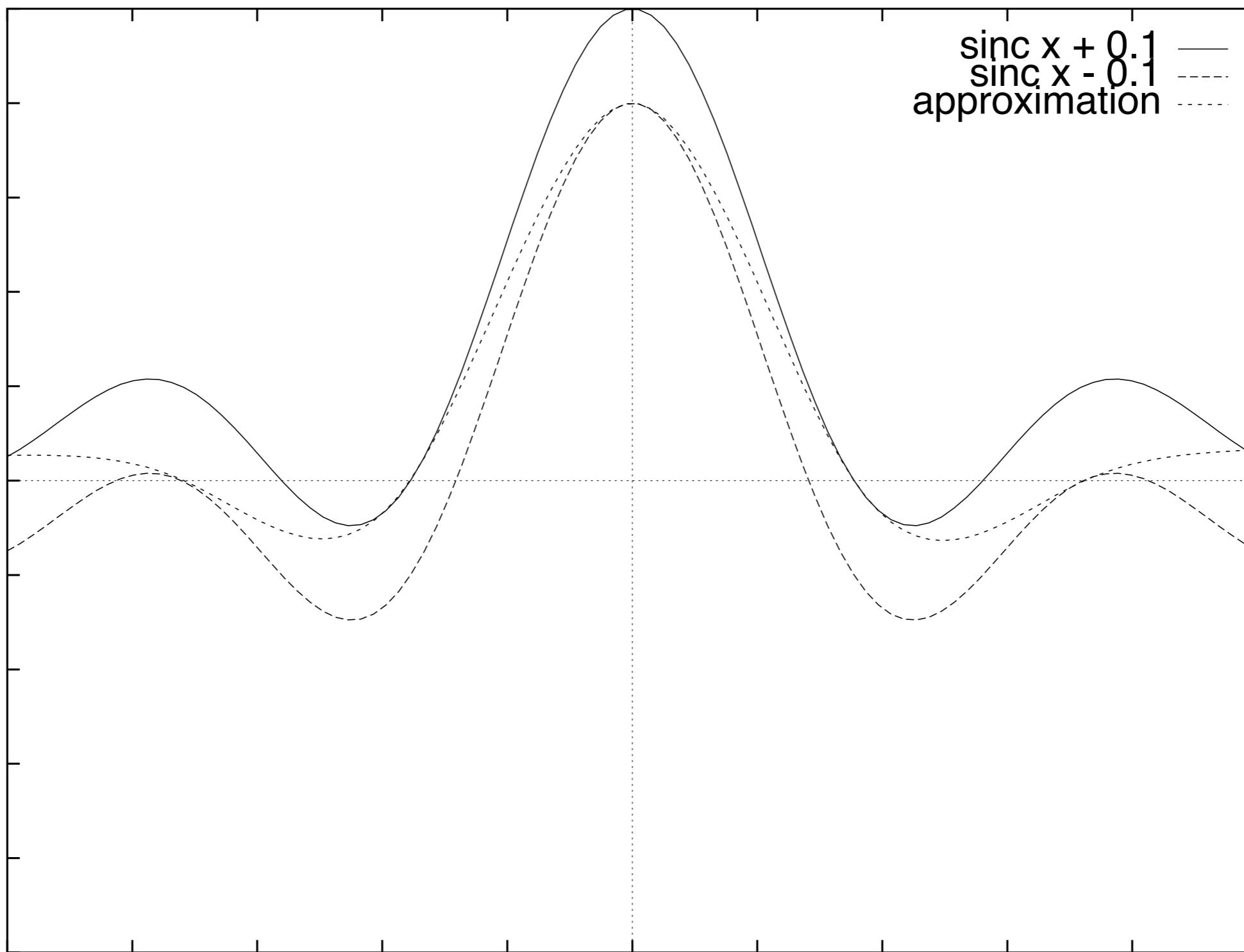
$$\underset{\alpha, \alpha^*}{\text{minimize}} \quad \frac{1}{2} (\alpha - \alpha^*)^\top K (\alpha - \alpha^*) + \epsilon \mathbf{1}^\top (\alpha + \alpha^*) + y^\top (\alpha - \alpha^*)$$

subject to $\mathbf{1}^\top (\alpha - \alpha^*) = 0$ and $\alpha_i, \alpha_i^* \in [0, C]$

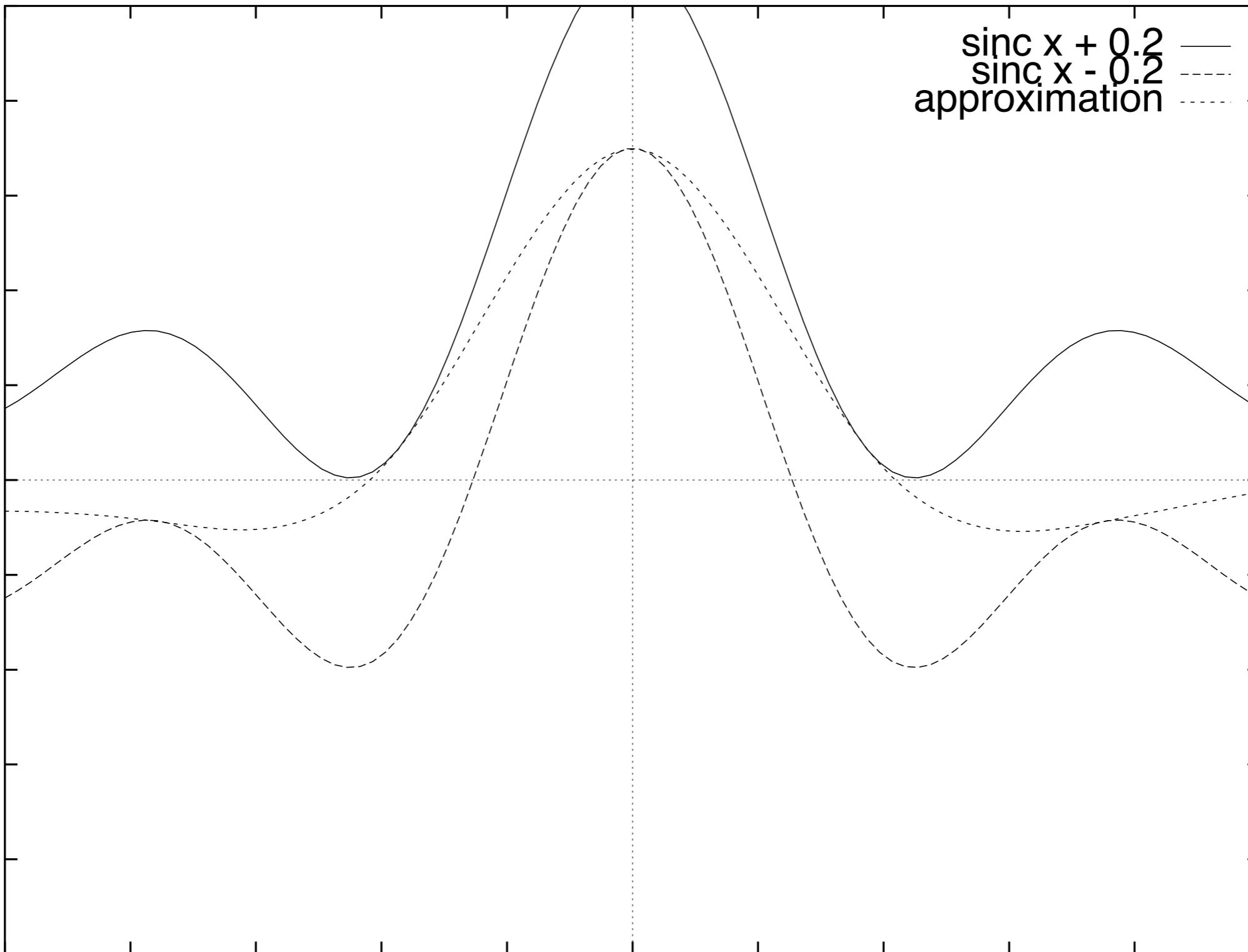
Properties

- Ignores ‘typical’ instances with small error
- Only upper or lower bound active at any time
- QP in $2n$ variables as cheap as SVM problem
- Robustness with respect to outliers
 - ℓ_1 loss yields same problem without epsilon
 - Huber’s robust loss yields similar problem but with added quadratic penalty on coefficients

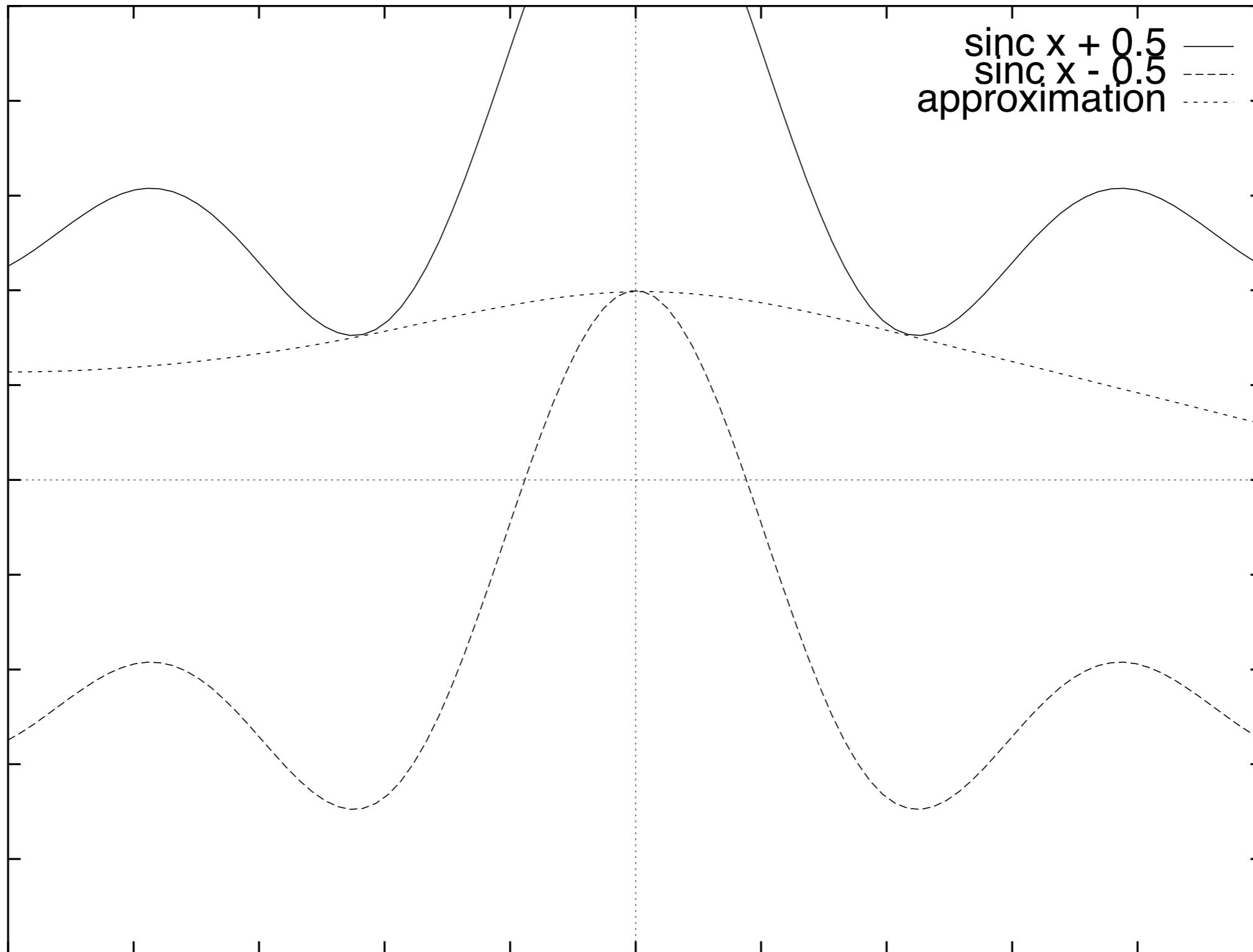
Regression example



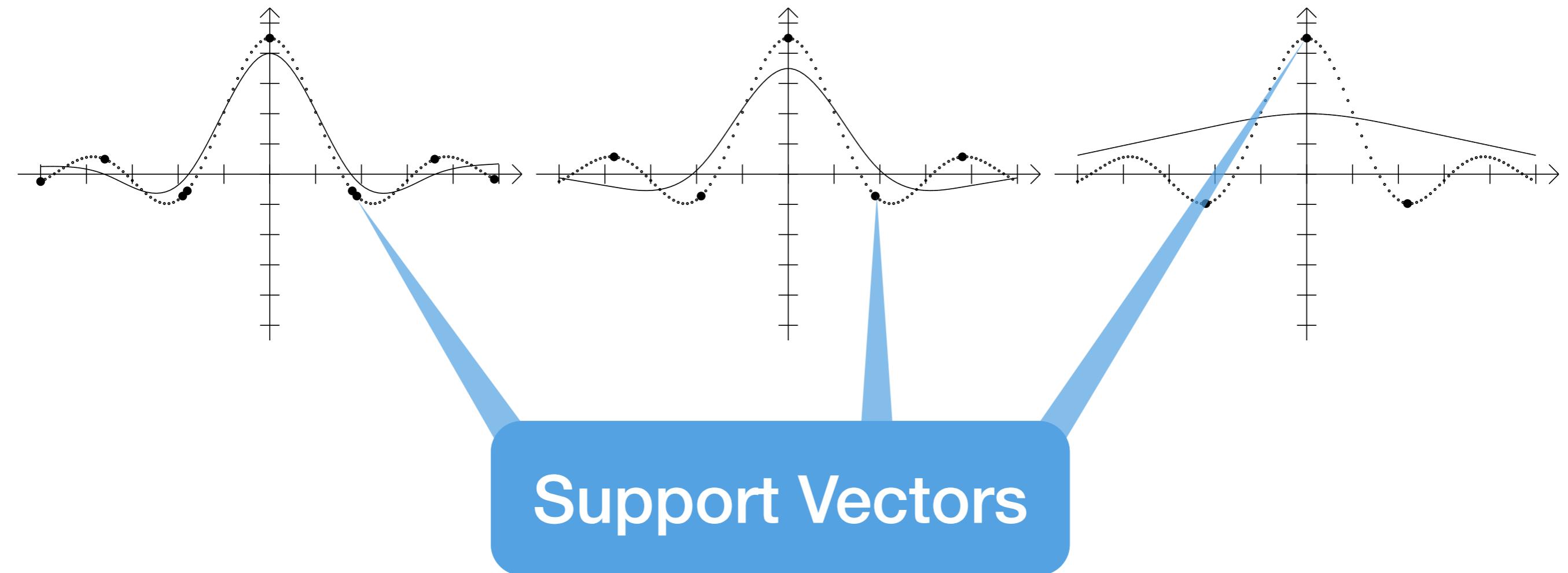
Regression example



Regression example



Regression example



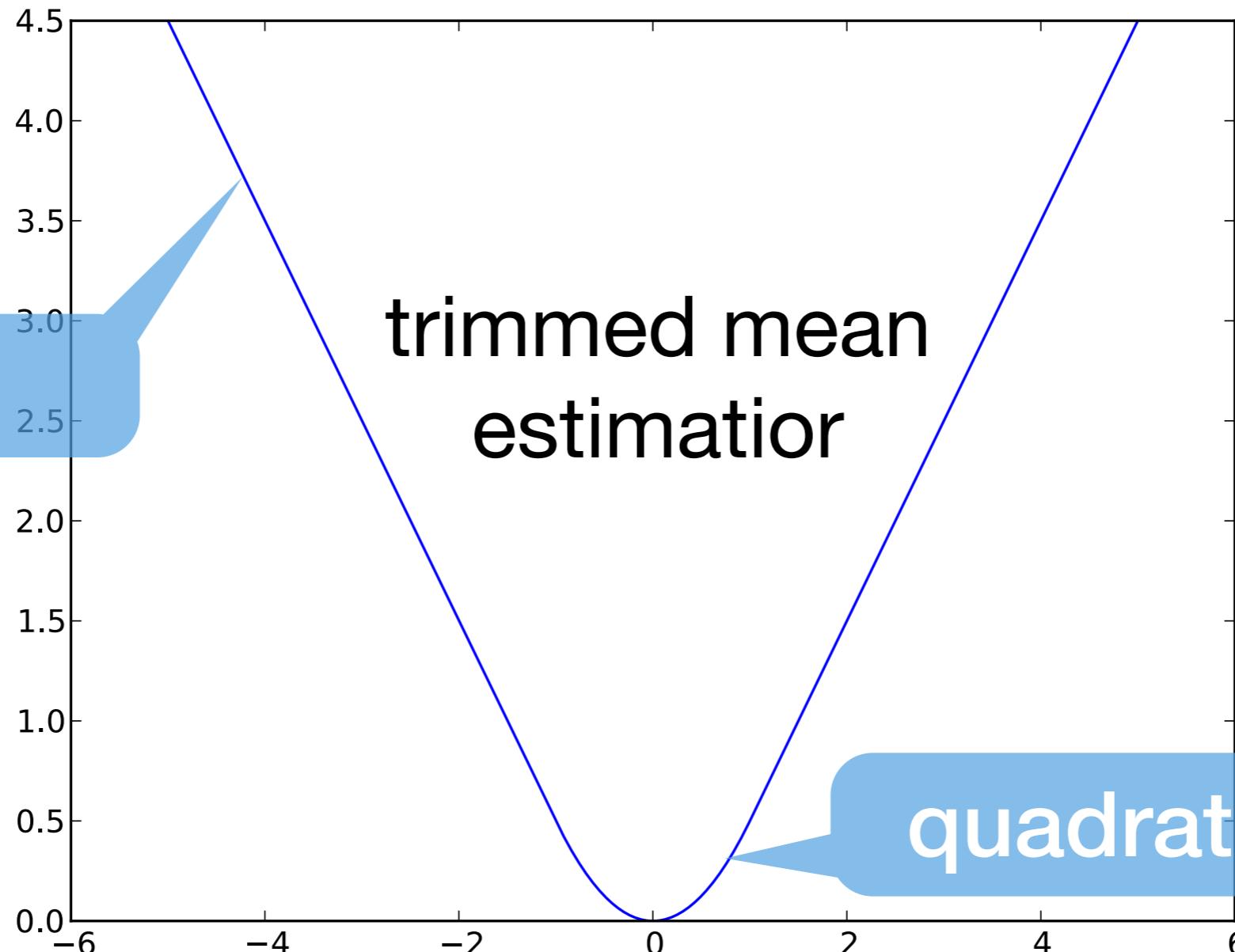
Huber's robust loss

$$l(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{if } |y - f(x)| < 1 \\ |y - f(x)| - \frac{1}{2} & \text{otherwise} \end{cases}$$

linear

trimmed mean
estimation

quadratic



Summary

- **Advantages:**
 - Kernels allow very flexible hypotheses
 - Poly-time exact optimization methods rather than approximate methods
 - Soft-margin extension permits mis-classified examples
 - Variable-sized hypothesis space
 - Excellent results (1.1% error rate on handwritten digits vs. LeNet's 0.9%)
- **Disadvantages:**
 - Must choose kernel parameters
 - Very large problems computationally intractable
 - Batch algorithm

Software

- **SVM^{light}**: one of the most widely used SVM packages.
Fast optimization, can handle very large datasets, C++ code.
- **LIBSVM**
- Both of these handle multi-class, weighted SVM for unbalanced data, etc.
- There are several new approaches to solving the SVM objective that can be much faster:
 - Stochastic subgradient method (discussed in a few lectures)
 - Distributed computation (also to be discussed)
- See <http://mloss.org>, “machine learning open source software”

Next Lecture:

Decision Trees