MACHINE LEARNING
DEPARTMENT

Learning with Kernels

10-715 Fall 2015

Alexander Smola
alex@smola.org

Office hours - after class & my office

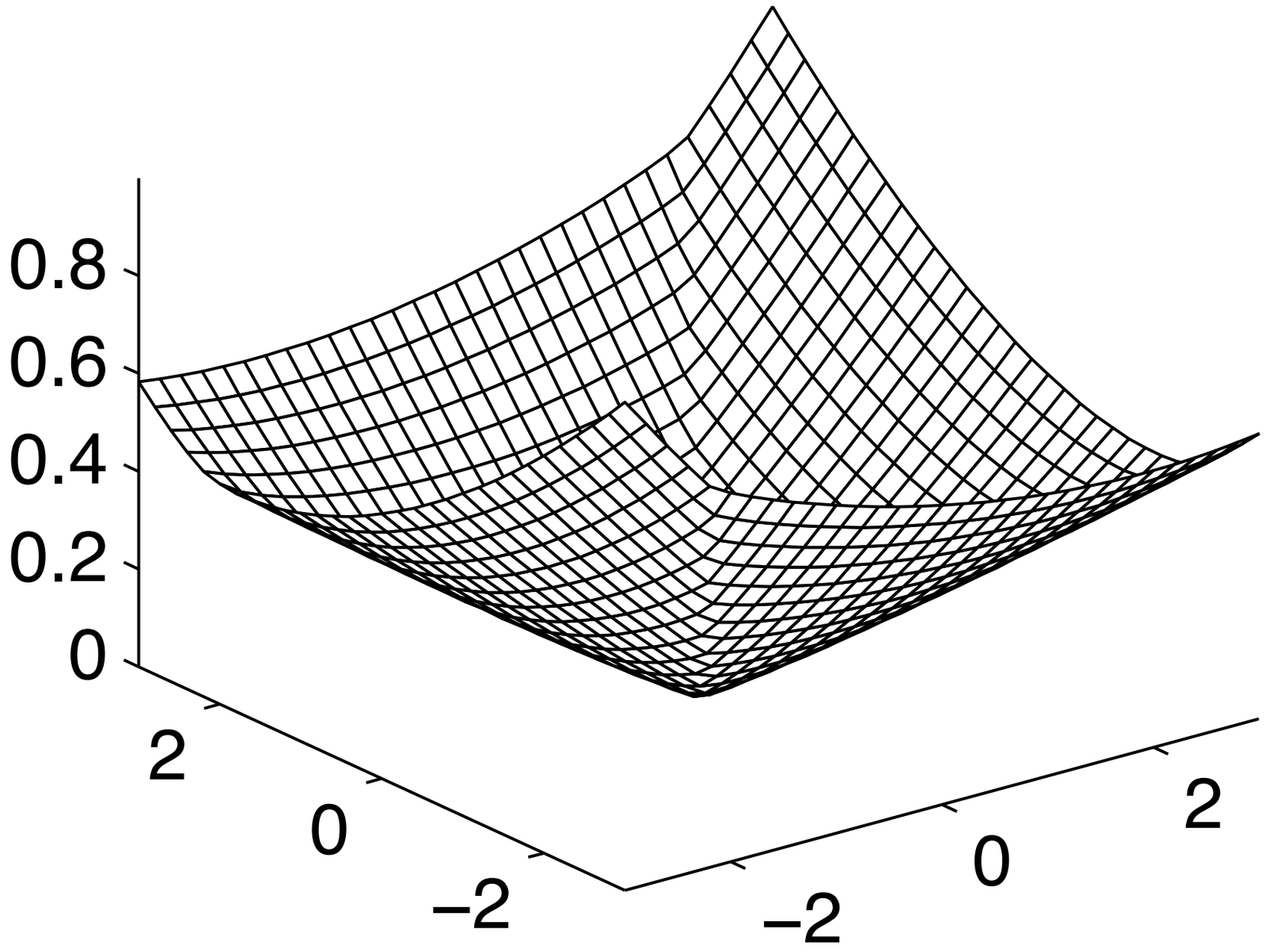
Marianas Labs

Carnegie Mellon University

Outline

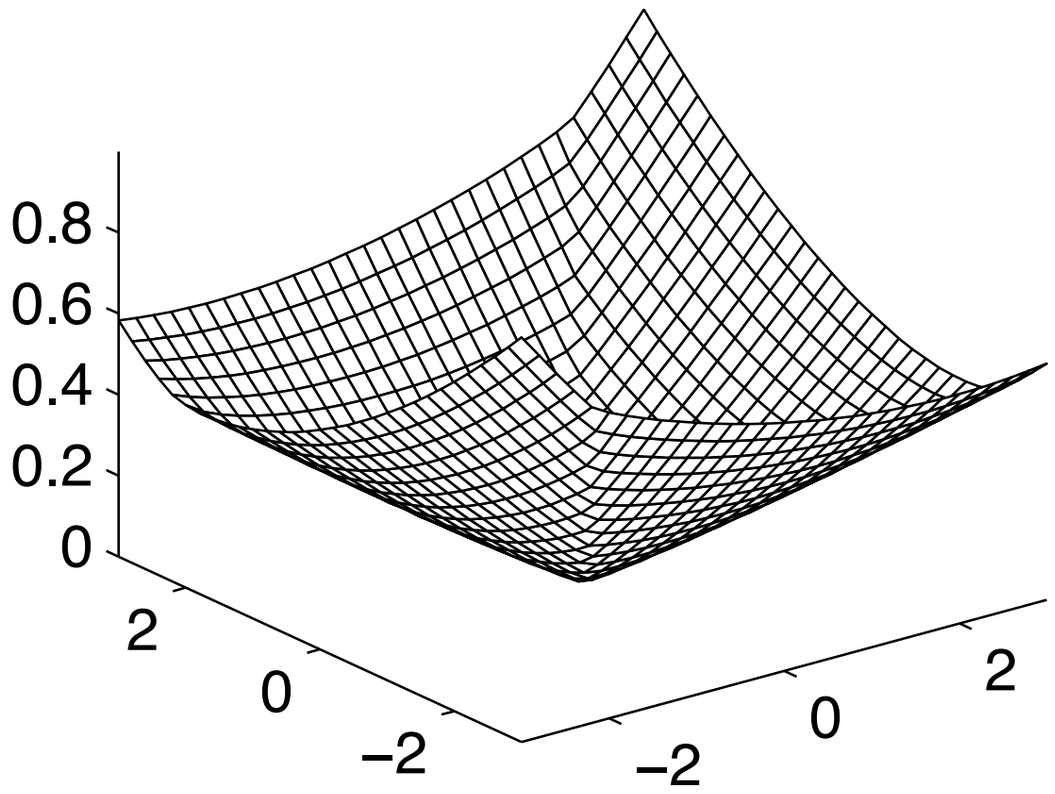
- **Convex Optimization**
  - Unconstrained Optimization
  - Constrained Optimization and Duality
  - Linear and Quadratic programs
- **Support Vector Machines**
  - Classification
  - Regression
  - Novelty Detection
- **Kernels**
  - Feature Space
  - Kernel PCA
  - Kernelized SVM

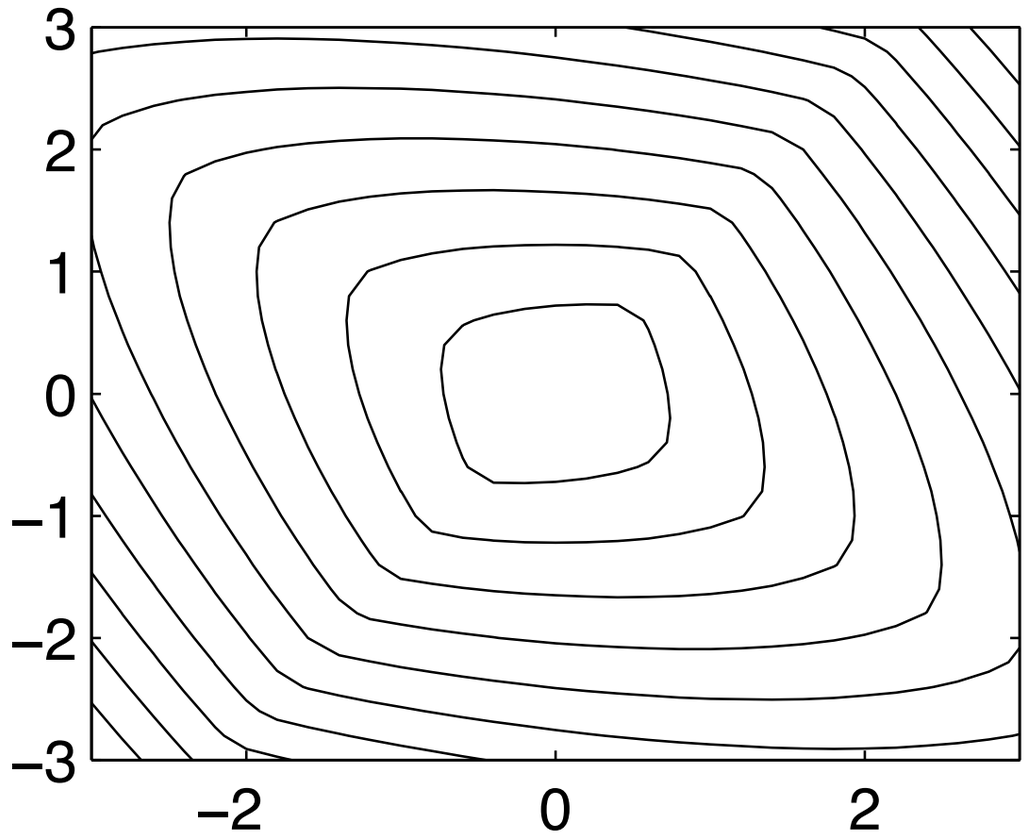
Unconstrained Optimization



Convexity 101

Convexity 101





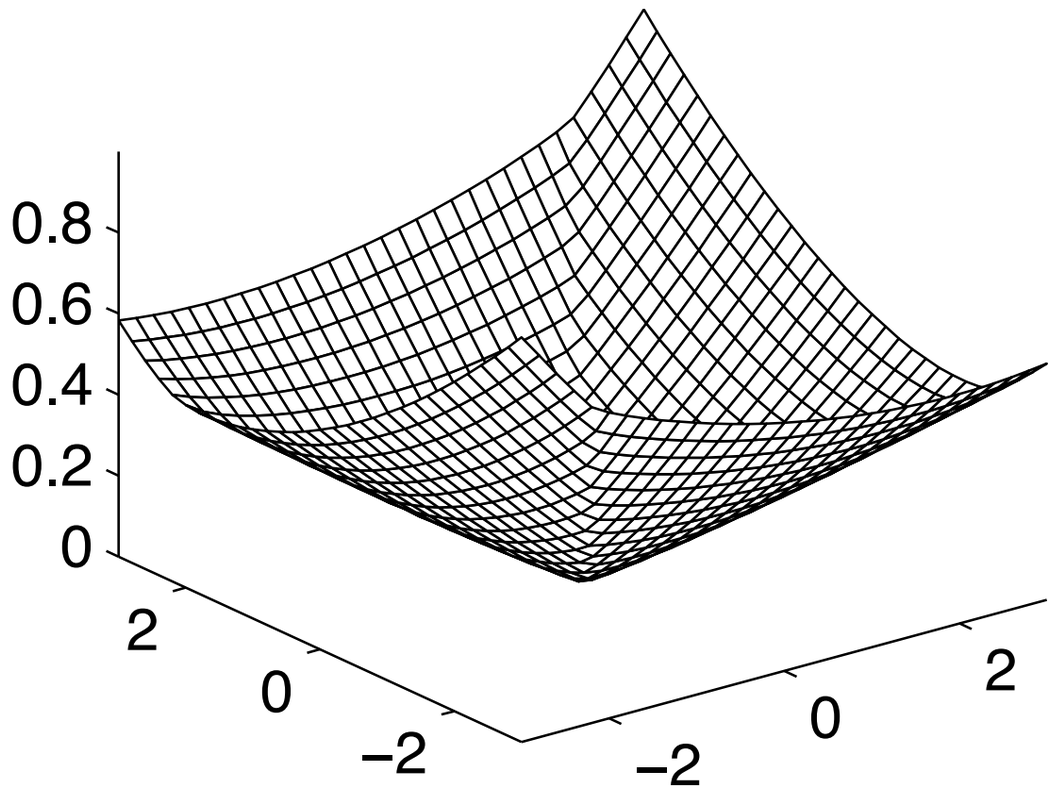
- Convex set

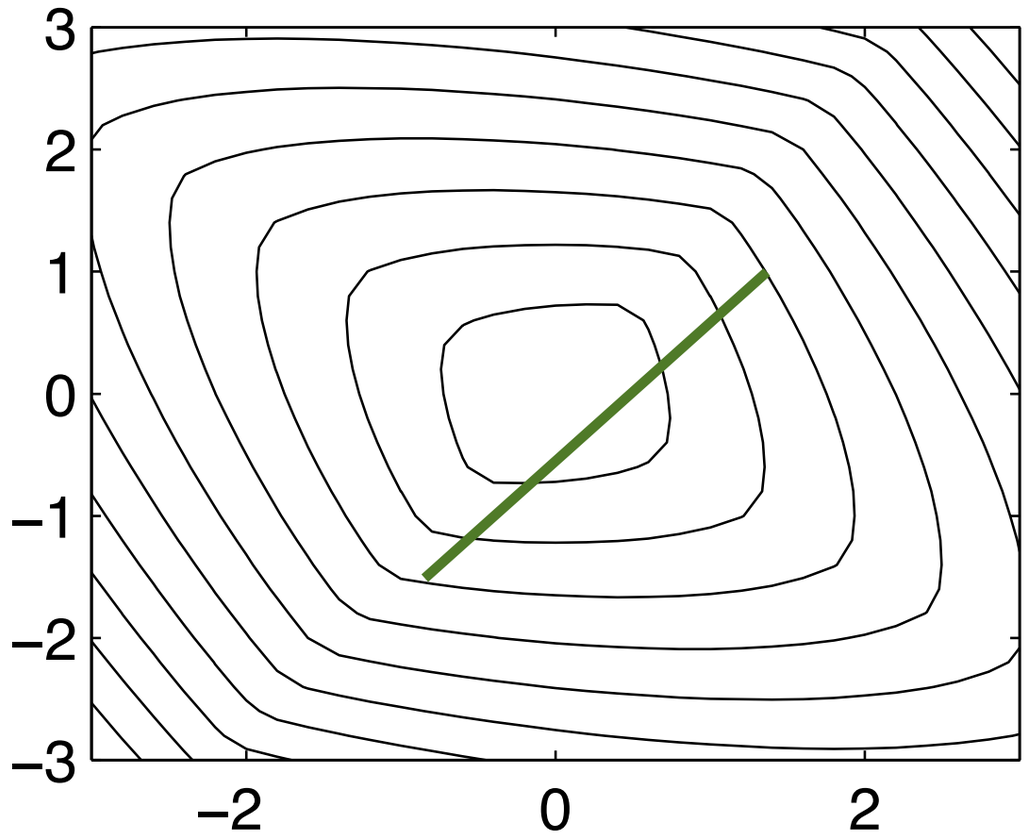
  For $x, x' \in X$ it follows that $\lambda x + (1-\lambda)x' \in X$ for $\lambda \in [0,1]$

- Convex function

$$\lambda\lambda f(x) + (1-\lambda) f(x') \geq f(\lambda x + (1-\lambda)x') \text{ for } \lambda \in [0,1]$$

Convexity 101





- Convex set

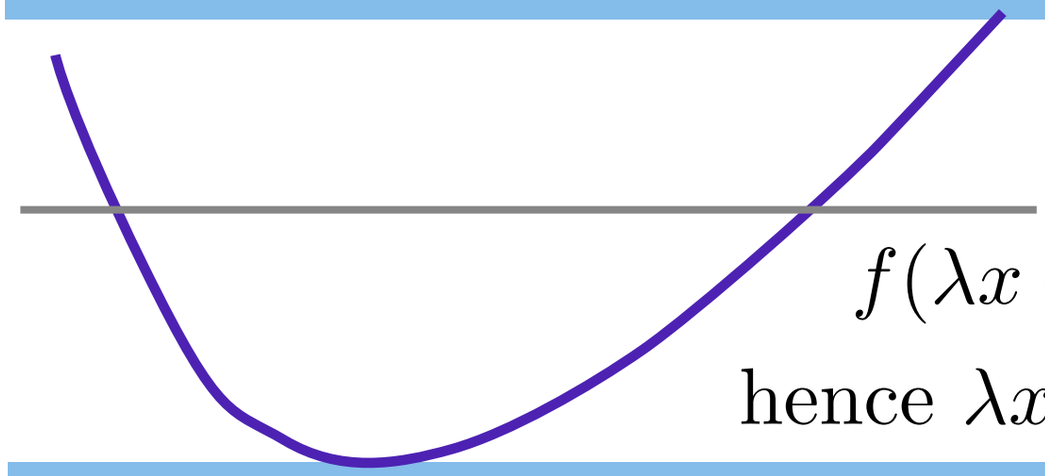
  For $x, x' \in X$ it follows that $\lambda x + (1-\lambda)x' \in X$ for $\lambda \in [0,1]$

- Convex function

$$\lambda\lambda f(x) + (1-\lambda) f(x') \geq f(\lambda x + (1-\lambda)x') \text{ for } \lambda \in [0,1]$$

Convexity 101

- Below-set of convex function is convex

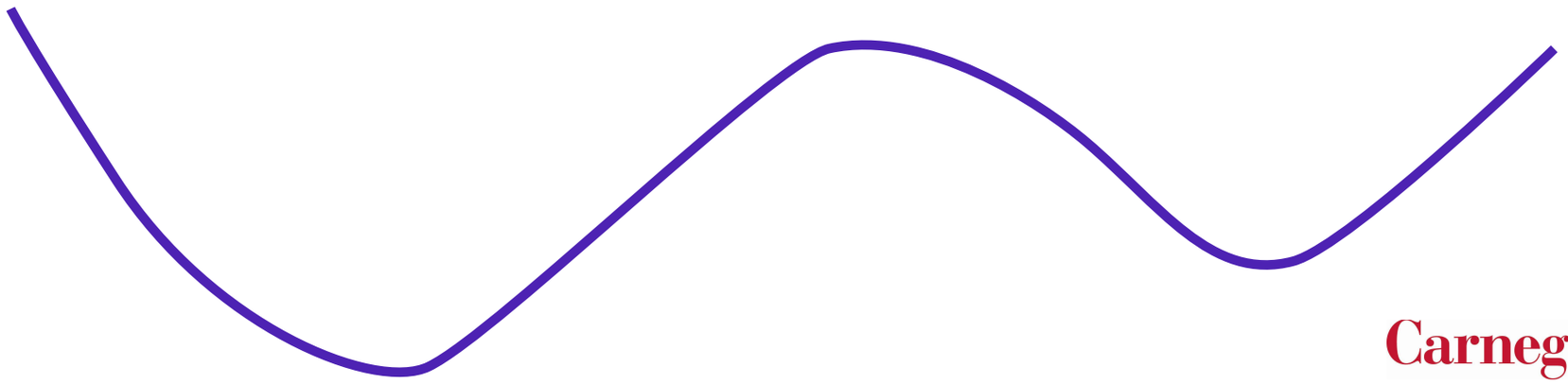


$$f(\lambda x + (1-\lambda)x') \leq \lambda f(x) + (1-\lambda) f(x')$$

$$\text{hence } \lambda x + (1-\lambda)x' \in X \text{ for } x, x' \in X$$

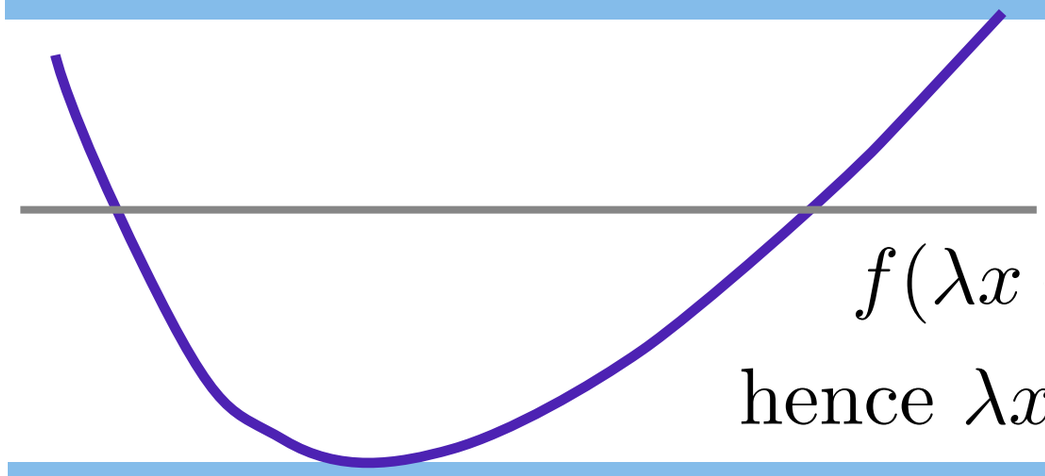
- Convex functions don't have local minima

  Proof by contradiction - linear interpolation breaks local minimum condition



Convexity 101

- Below-set of convex function is convex

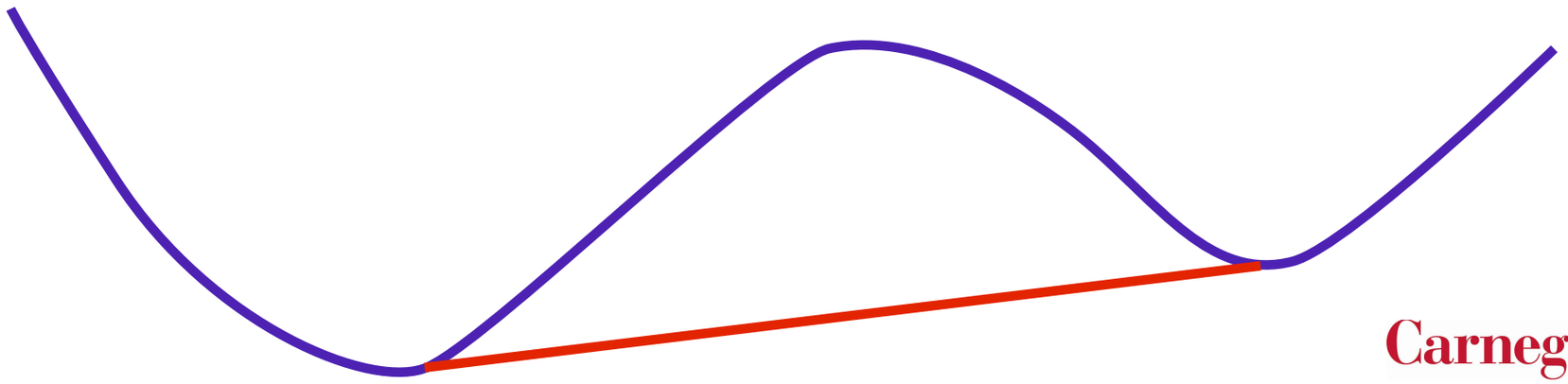


$$f(\lambda x + (1-\lambda)x') \leq \lambda f(x) + (1-\lambda) f(x')$$

$$\text{hence } \lambda x + (1-\lambda)x' \in X \text{ for } x, x' \in X$$

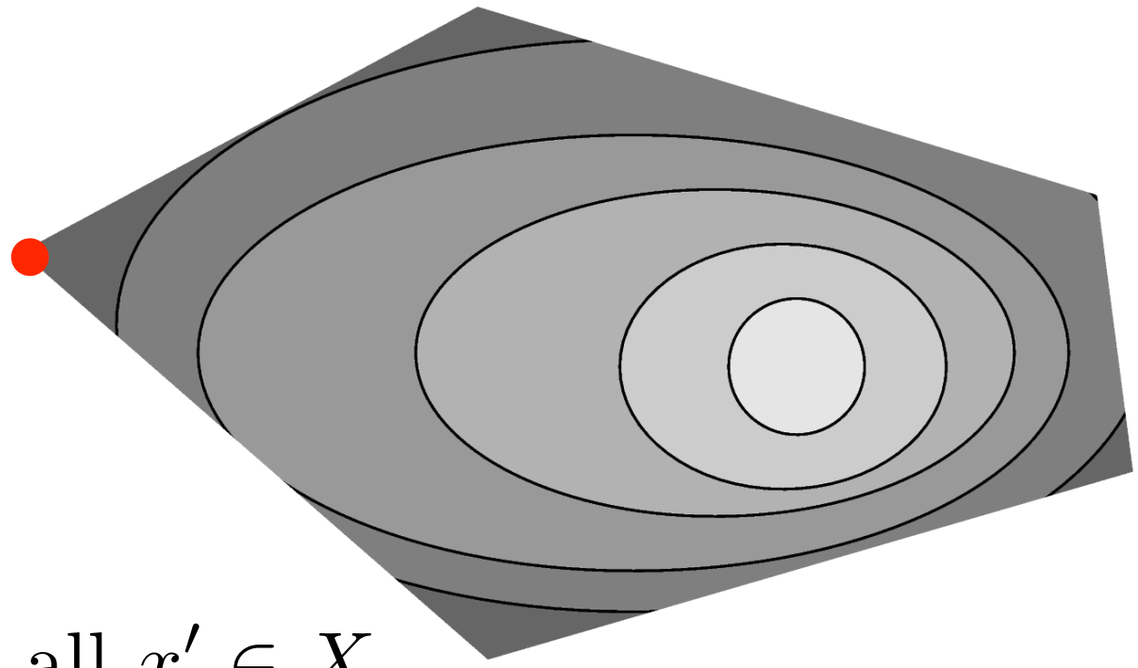
- Convex functions don't have local minima

  Proof by contradiction - linear interpolation breaks local minimum condition



Convexity 101

- Vertex of a convex set
  Point which cannot be extrapolated within convex set



$$\lambda x + (1-\lambda)x' \notin X \text{ for } \lambda > 1 \text{ for all } x' \in X$$

- Convex hull

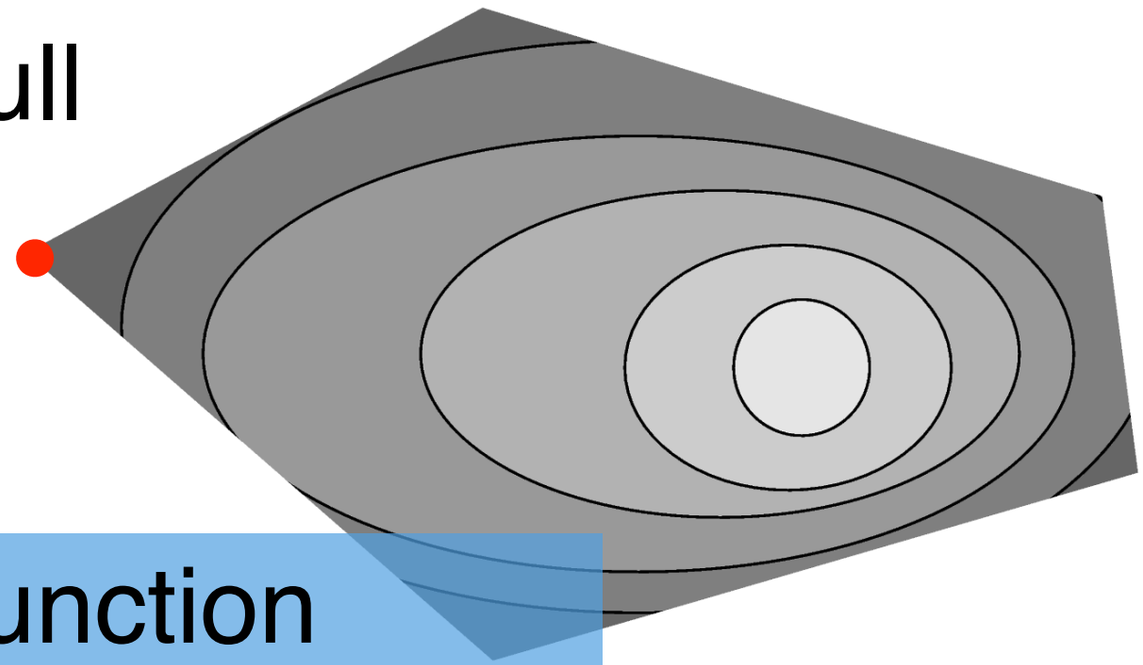
$$\operatorname{co} X := \left\{ \bar{x} \,\middle|\, \bar{x} = \sum_{i=1}^{n} \alpha_i x_i \;\; where \; n \in \mathbb{N}, \alpha_i \geq 0 \;\; and \sum_{i=1}^{n} \alpha_i \leq 1 \right\}$$

- Convex hull of set is a convex set

Convexity 101

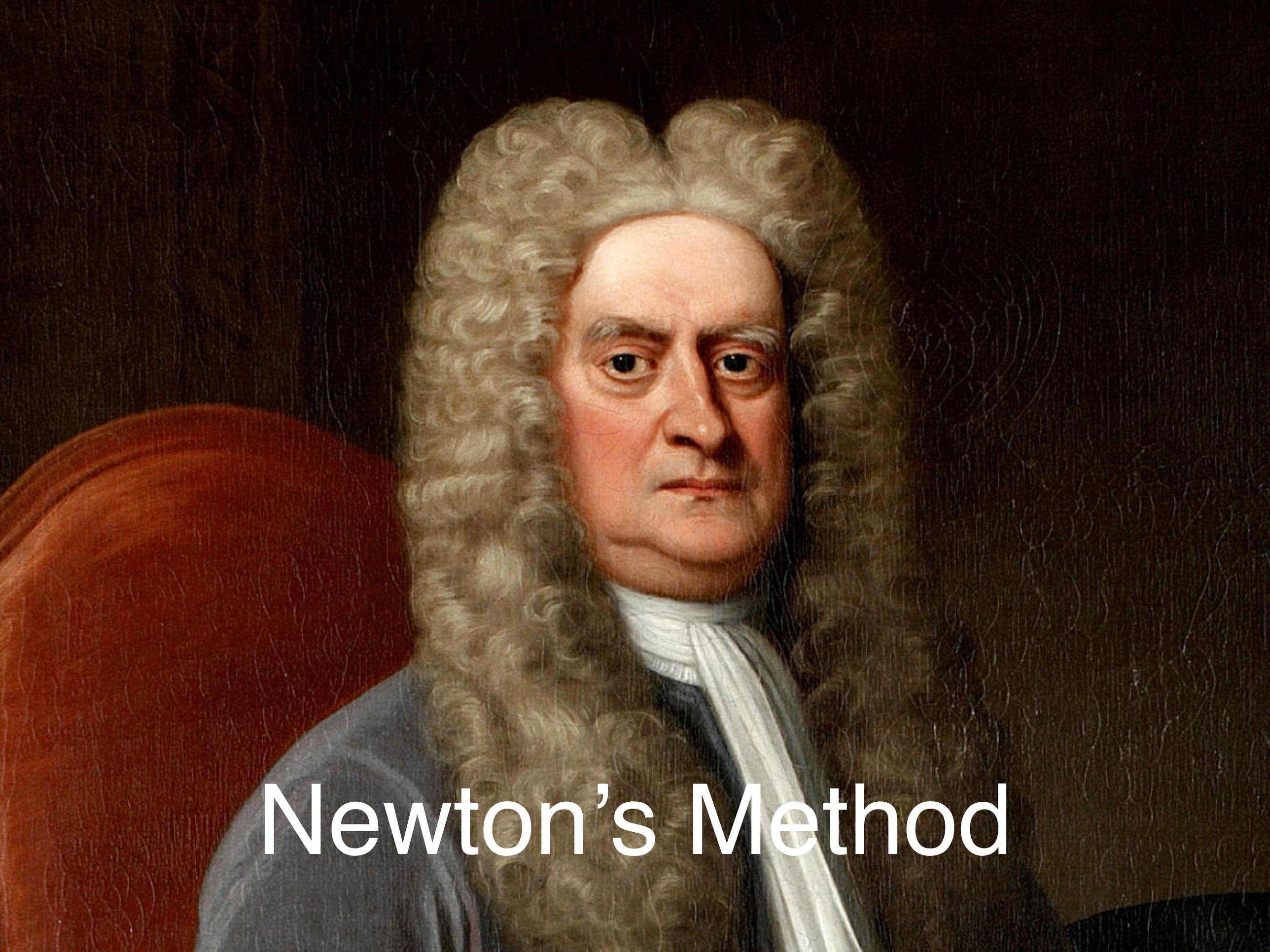
- Supremum on convex hull

$$\sup_{x \in X} f(x) = \sup_{x \in \mathrm{co} X} f(x)$$



Proof by contradiction

- Maximum over convex function on convex set is obtained on vertex
  - Assume that maximum inside line segment
  - Then function cannot be convex
  - Hence it must be on vertex



Newton's Method

Newton Method

- Convex objective function f
- Nonnegative second derivative

$$\partial_x^2 f(x) \succeq 0$$

- Taylor expansion

$$f(x+\delta) = f(x) + \langle \delta, \partial_x f(x) \rangle + \frac{1}{2}\delta^\top \partial_x^2 f(x)\delta + O(\delta^3)$$

Hessian

gradient

- Minimize approximation & iterate until converged

$$x \leftarrow x - \left[\partial_x^2 f(x)\right]^{-1} \partial_x f(x)$$

Convergence Analysis

- There exists a region around optimality where Newton's method converges quadratically if f is twice continuously differentiable
- For some region around x* gradient is well approximated by Taylor expansion

$$\left\| \partial_x f(x^*) - \partial_x f(x) - \left\langle x^* - x, \partial_x^2 f(x) \right\rangle \right\| \leq \gamma \left\| x^* - x \right\|^2$$

- Expand Newton update

$$\begin{aligned} \|x_{n+1} - x^*\| &= \left\| x_n - x^* - \left[\partial_x^2 f(x_n)\right]^{-1} \left[\partial_x f(x_n) - \partial_x f(x^*)\right] \right\| \\ &= \left\| \left[\partial_x^2 f(x_n)\right]^{-1} \left[\partial_x^f (x_n)[x_n - x^*] - \partial_x f(x_n) + \partial_x f(x^*)\right] \right\| \\ &\leq \gamma \left\| \left[\partial_x^2 f(x_n)\right]^{-1} \right\| \|x_n - x^*\|^2 \end{aligned}$$

Convergence Analysis

- Two convergence regimes
  - As slow as gradient descent outside the region where Taylor expansion is good

$$\left\|\partial_x f(x^*) - \partial_x f(x) - \left\langle x^* - x, \partial_x^2 f(x)\right\rangle\right\| \leq \gamma \left\|x^* - x\right\|^2$$

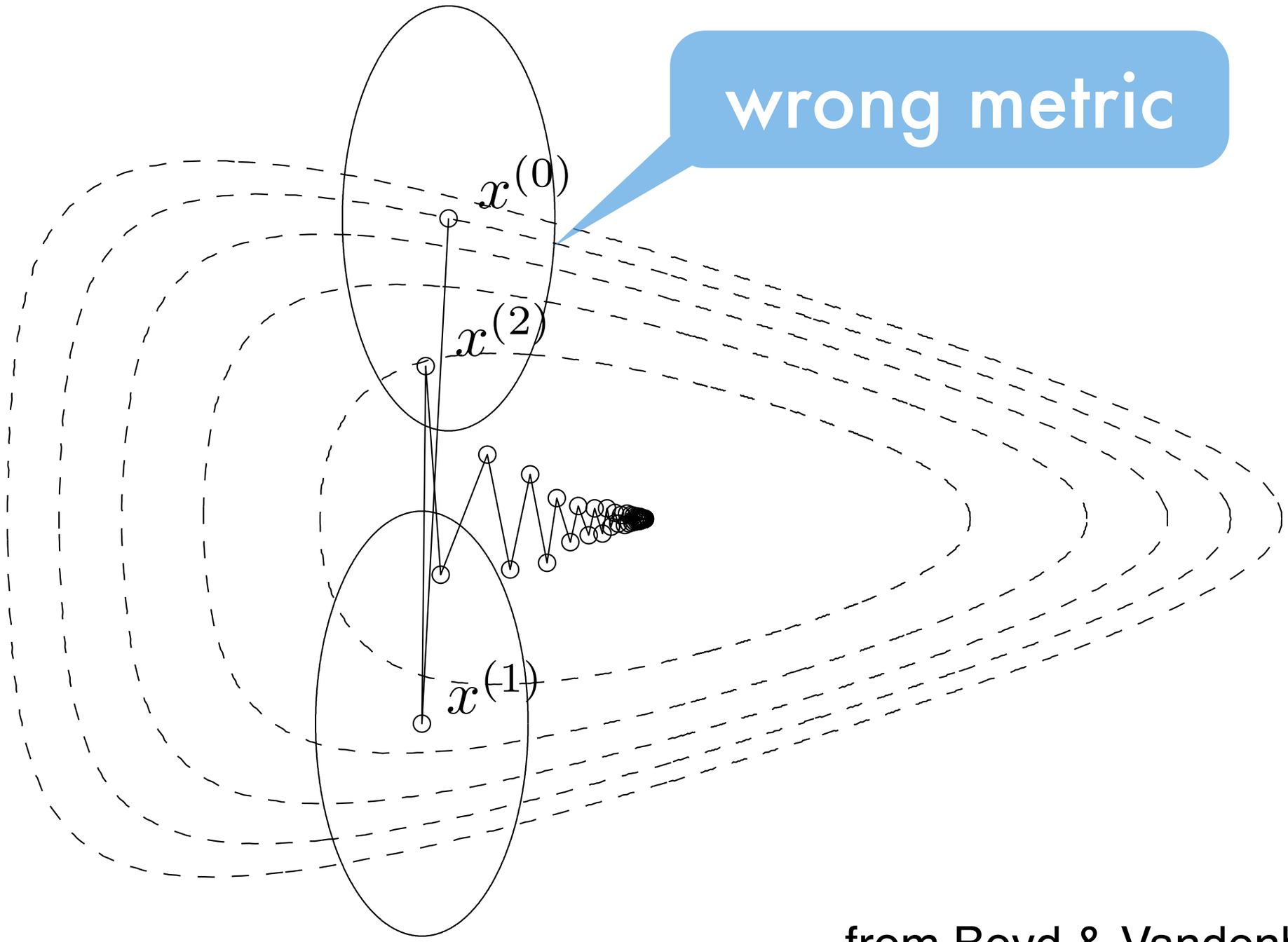
- Quadratic convergence once the bound holds

$$\|x_{n+1} - x^*\| \leq \gamma \left\|\left[\partial_x^2 f(x_n)\right]^{-1}\right\| \|x_n - x^*\|^2$$

- Newton method is affine invariant (proof by chain rule)

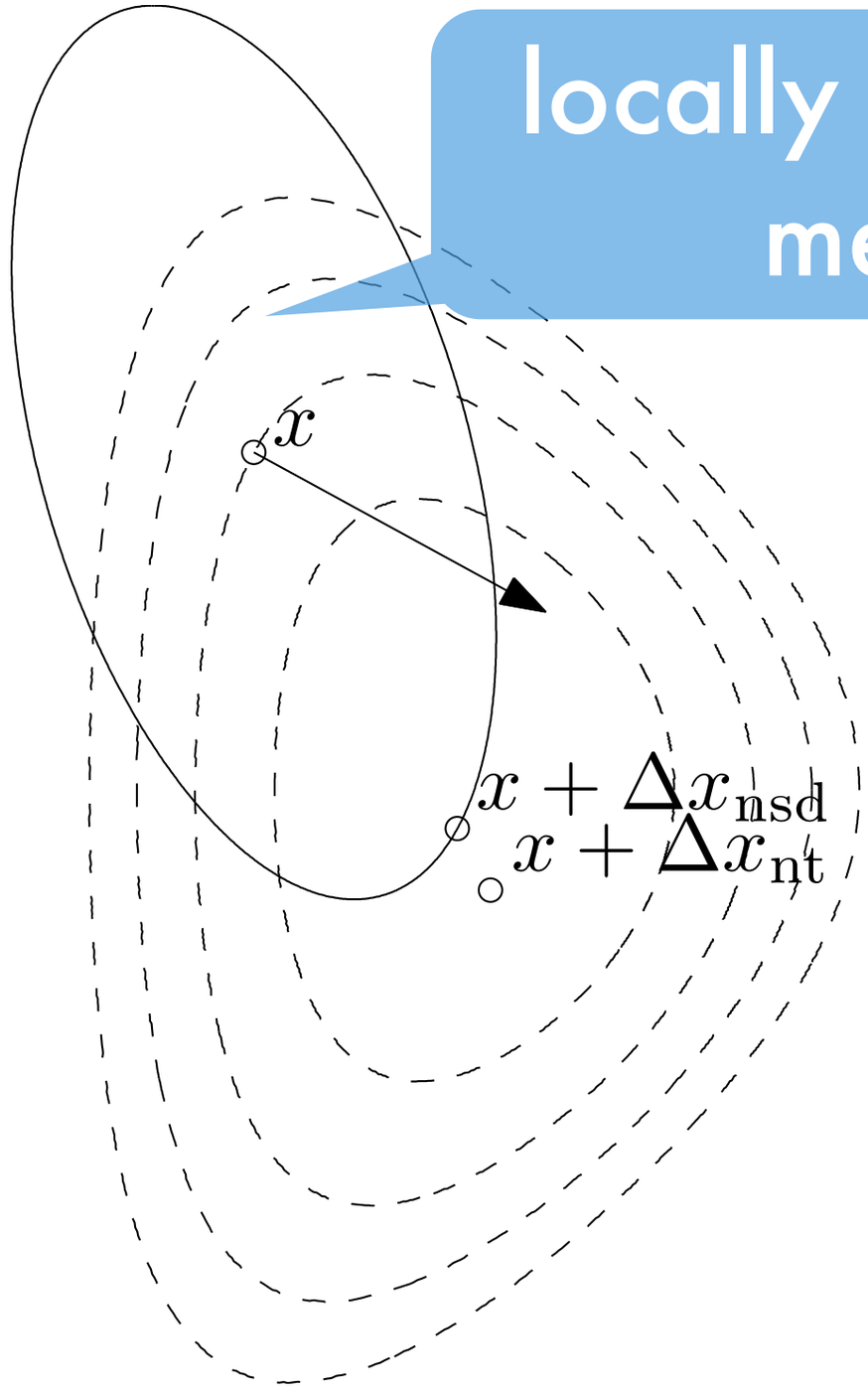
See Boyd and Vandenberghe, Chapter 9.5 for much more

Newton method rescales space



from Boyd & Vandenberghe

Newton method rescales space

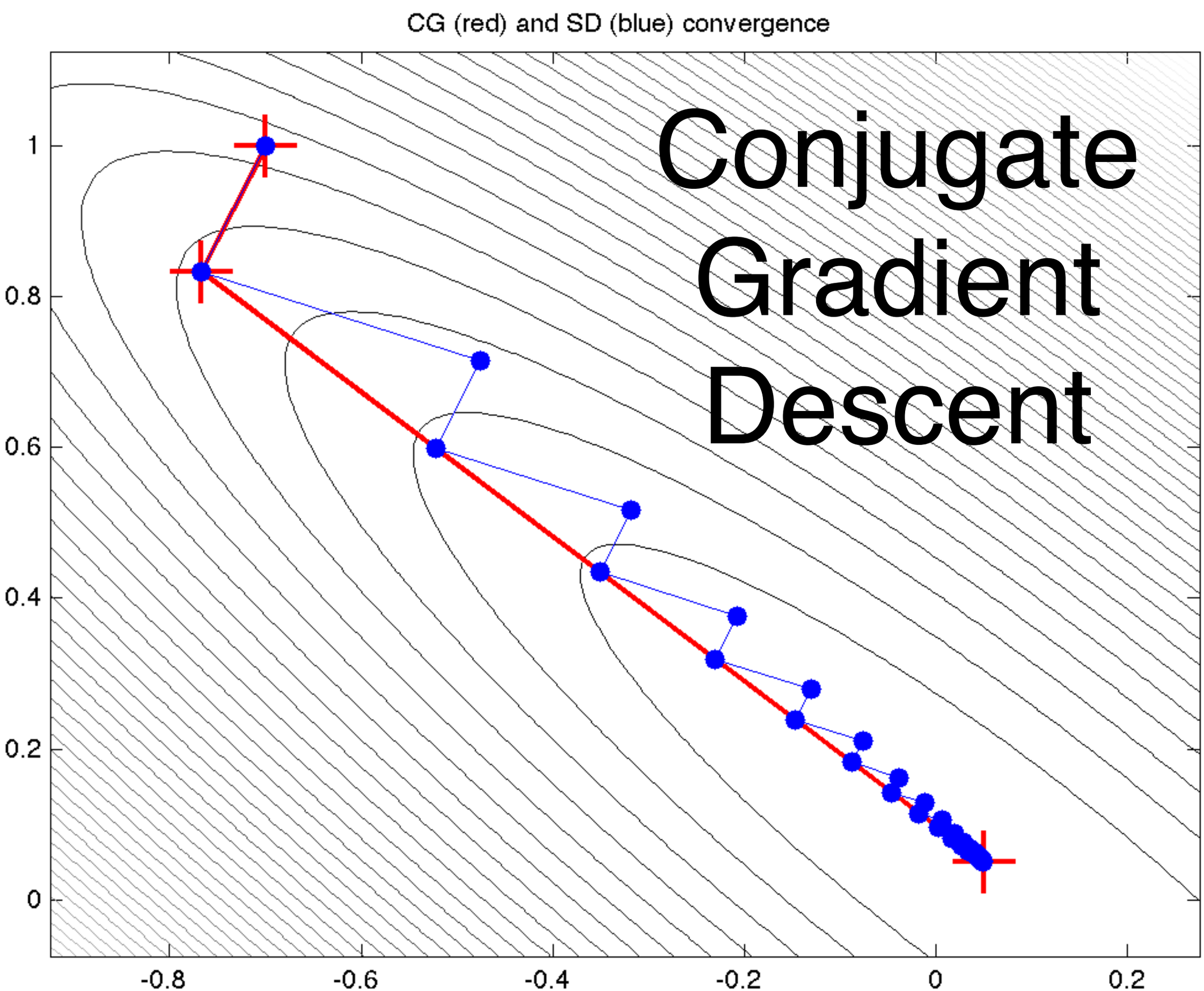


Boyd & Vandenberghe

Carnegie Mellon University

Parallel Newton Method

- Good rate of convergence
- Few passes through data needed
- Parallel aggregation of gradient and Hessian
- Gradient requires $O(d)$ data
- Hessian requires $O(d^2)$ data
- Update step is $O(d^3)$ & nontrivial to parallelize
- Use it only for low dimensional problems


CG (red) and SD (blue) convergence
1
0.8
0.6
0.4
0.2
0
-0.8
-0.6
-0.4
-0.2
0
0.2
Conjugate
Gradient
Descent

Key Idea

- Minimizing quadratic function $(K \succeq 0)$

$$f(x) = \frac{1}{2} x^\top K x - l^\top x + c$$

  takes cubic time (e.g. Cholesky factorization)
- Matrix vector products and orthogonalization

- Vectors x, x' are K orthogonal if $x^\top K x' = 0$
- m mutually K orthogonal vectors $x_i \in \mathbb{R}^m$
  - form a basis
  - allow expansion
  - solve linear system

$$z = \sum_{i=1}^{m} x_i \frac{x_i^\top K z}{x_i^\top K x_i}$$

$$z = \sum_{i=1}^{m} x_i \frac{x_i^\top y}{x_i^\top K x_i} \text{ for } y = Kz$$

Proof

- m mutually K orthogonal vectors $x_i \in \mathbb{R}^m$
  - form a basis
  - allow expansion
  - solve linear system

$$z = \sum_{i=1}^{m} x_i \frac{x_i^\top K z}{x_i^\top K x_i}$$

$$z = \sum_{i=1}^{m} x_i \frac{x_i^\top y}{x_i^\top K x_i} \text{ for } y = Kz$$

- Show linear independence by contradiction

$$\sum_i \alpha_i x_i = 0 \text{ hence } 0 = x_j^\top K \sum_i \alpha_i x_i = x_j^\top K x_j \alpha_j$$

- Reconstruction - expand z into basis

$$z = \sum_i \alpha_i x_i \text{ hence } x_j^\top K z = x_j^\top K \sum_i \alpha_i x_i = x_j^\top K x_j \alpha_j$$

- For linear system plug in y = Kz

Conjugate Gradient Descent

- Gradient computation

$$f(x) = \frac{1}{2} x^\top K x - l^\top x + c \text{ hence } g(x) = Kx - l$$

- Algorithm

initialize x_0 and $v_0 = g_0 = Kx_0 - l$ and $i = 0$

repeat

$x_{i+1} = x_i - v_i \frac{g_i^\top v_i}{v_i^\top K v_i}$

$g_{i+1} = Kx_{i+1} - l$

$v_{i+1} = -g_{i+1} + v_i \frac{g_{i+1}^\top K v_i}{v_i^\top K v_i}$

$i \leftarrow i + 1$

until $g_i = 0$

deflation step

K orthogonal

Proof - Deflation property

$$x_{i+1} = x_i - v_i \frac{g_i^\top v_i}{v_i^\top K v_i}$$
$$g_{i+1} = K x_{i+1} - l$$
$$v_{i+1} = -g_{i+1} + v_i \frac{g_{i+1}^\top K v_i}{v_i^\top K v_i}$$

- First assume that the vi are K orthogonal and show that xi+1 is optimal in span of {v1 .. vi}
- Enough if we show that $v_j^\top g_i = 0 \text{ for all } j < i$
  - For j=i expand

$$v_i^\top g_{i+1} = v_i^\top \left[K x_i - l - K v_i \frac{g_i^\top v_i}{v_i^\top K v_i} \right]$$
$$= v_i^\top g_i - v_i^\top K v_i \frac{g_i^\top v_i}{v_i^\top K v_i} = 0$$

  - For smaller j a consequence of K orthogonality

Proof - K orthogonality

$$x_{i+1} = x_i - v_i \frac{g_i^\top v_i}{v_i^\top K v_i}$$
$$g_{i+1} = K x_{i+1} - l$$
$$v_{i+1} = -g_{i+1} + v_i \frac{g_{i+1}^\top K v_i}{v_i^\top K v_i}$$

- Need to check that vi+1 is K orthogonal to all vj (rest automatically true by construction)

$$v_j^\top K v_{i+1} = -v_j^\top K g_{i+1} + v_j^\top K v_i \frac{g_{i+1}^\top K v_i}{v_i^\top K V_i}$$

0 by deflation

0 by K orthogonality

Properties

- Subspace expansion method for optimality ($g, Kg, K^2g, K^3g, ...$)
- Focuses on leading eigenvalues
- Often sufficient to take only a few steps (whenever the eigenvalues decay rapidly)

Extensions

Generic Method	Compute Hessian $K_i := f''(x_i)$ and update α_i, β_i with $\alpha_i = -\frac{g_i^\top v_i}{v_i^\top K_i v_i}$ $\beta_i = \frac{g_{i+1}^\top K_i v_i}{v_i^\top K_i v_i}$ This requires calculation of the Hessian at each iteration.
Fletcher–Reeves [163]	Find α_i via a line search and use Theorem 6.20 (iii) for β_i $\alpha_i = \operatorname{argmin}_\alpha f(x_i + \alpha v_i)$ $\beta_i = \frac{g_{i+1}^\top g_{i+1}}{g_i^\top g_i}$
Polak–Ribiere [398]	Find α_i via a line search $\alpha_i = \operatorname{argmin}_\alpha f(x_i + \alpha v_i)$ $\beta_i = \frac{(g_{i+1}-g_i)^\top g_{i+1}}{g_i^\top g_i}$ Experimentally, Polak–Ribiere tends to be better than Fletcher–Reeves.

x and v updates

Broyden-Fletcher-Goldfarb-Shanno

Basic Idea

- Newton-like method to compute descent direction

$$\delta_i = B_i^{-1} \partial_x f(x_{i-1})$$

- Line search on f in direction

$$x_{i+1} = x_i - \alpha_i \delta_i$$

- Update B with rank 2 matrix

$$B_{i+1} = B_i + u_i u_i^\top + v_i v_i^\top$$

- Require that Quasi-Newton condition holds

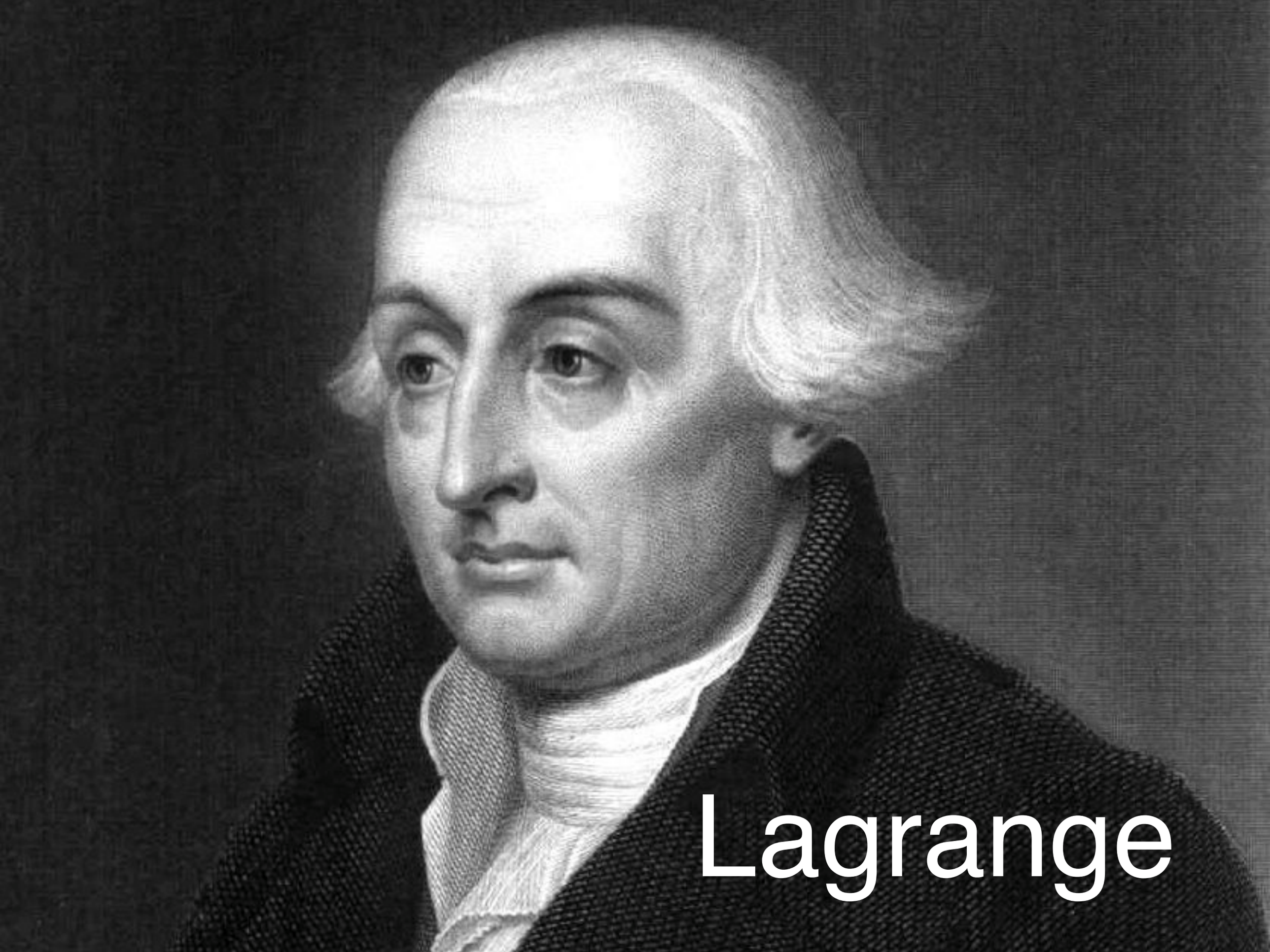
$$B_{i+1}(x_{i+1} - x_i) = \partial_x f(x_{i+1}) - \partial_x f(x_i)$$

$$B_{i+1} = B_i + \frac{g_i g_i^\top}{\alpha_i \delta_i^\top g_i} - \frac{B_i \delta_i \delta_i^\top B_i}{\delta_i^\top B_i \delta_i}$$

Properties

- Simple rank 2 update for B
- Use matrix inversion lemma to update inverse
- Memory-limited versions L-BFGS
- Use toolbox if possible (TAO, MATLAB) (typically slower if you implement it yourself)
- Works well for nonlinear nonconvex objectives (often even for nonsmooth objectives)

Constrained Convex Optimization

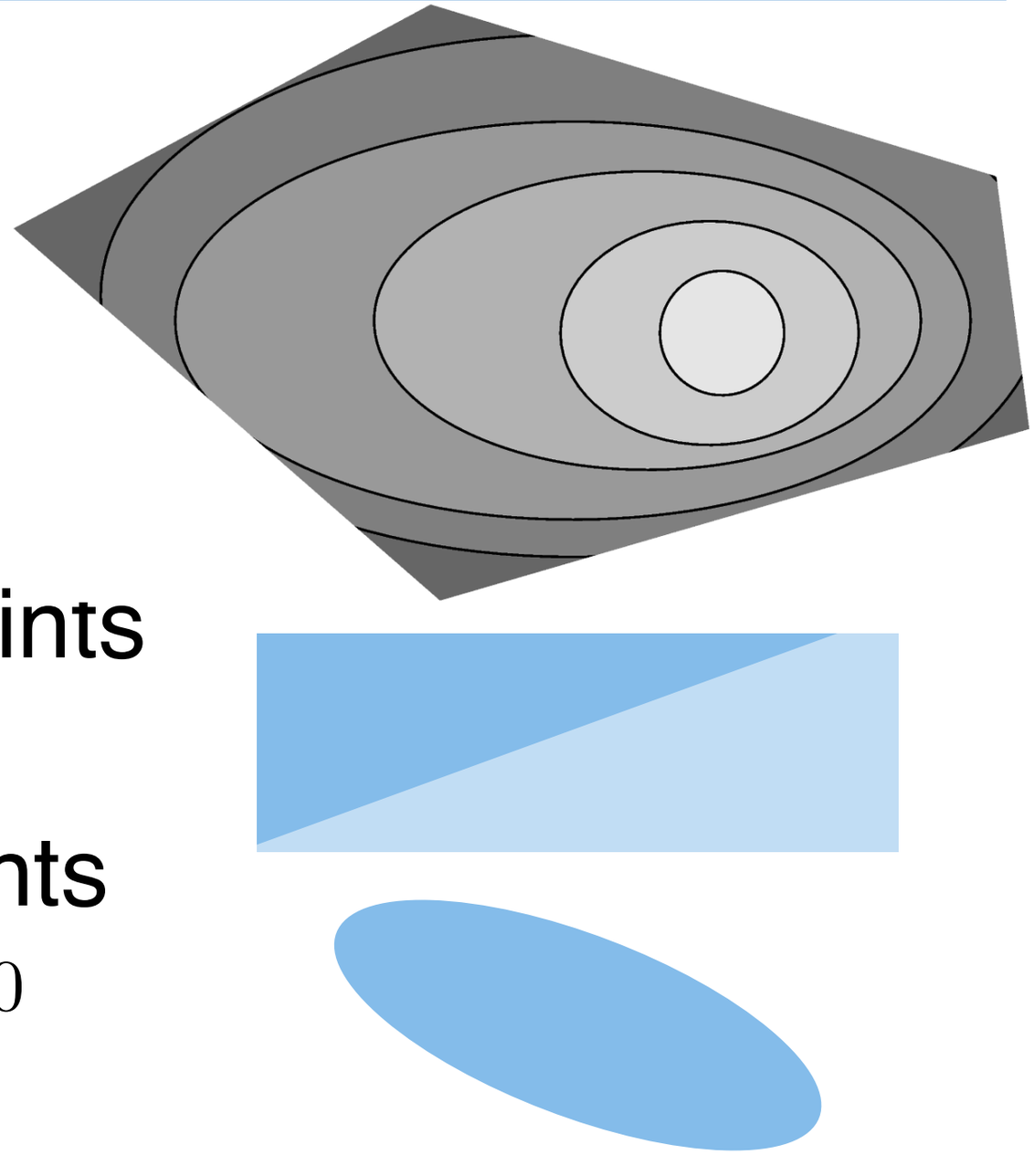

Lagrange

Constrained Convex Minimization

- Optimization problem

$$\underset{x}{\text{minimize}}\ f(x)$$

$$\text{subject to } c_i(x) \leq 0 \text{ for all } i$$



- Common constraints
  - linear inequality constraints

$$\langle w_i, x \rangle + b_i \leq 0$$

  - quadratic cone constraints

$$x^\top Q x + b^\top x \leq c \text{ with } Q \succeq 0$$

  - semidefinite constraints

$$M \succeq 0 \text{ or } M_0 + \sum_i x_i M_i \succeq 0$$

Constrained Convex Minimization

- Optimization problem

$$\underset{x}{\text{minimize}}\ f(x)$$

$$\text{subject to } c_i(x) \leq 0 \text{ for}$$

- Common constrain
  - linear inequality constraints

$$\langle w_i, x \rangle + b_i \leq 0$$

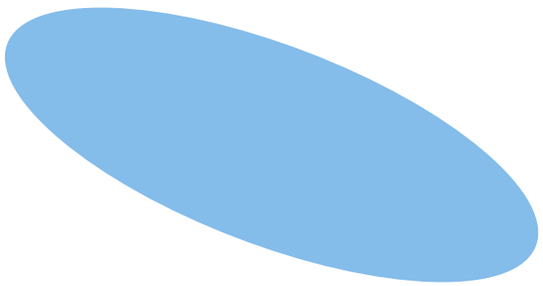
  - quadratic cone constraints

$$x^\top Q x + b^\top x \leq c \text{ with } Q \succeq 0$$

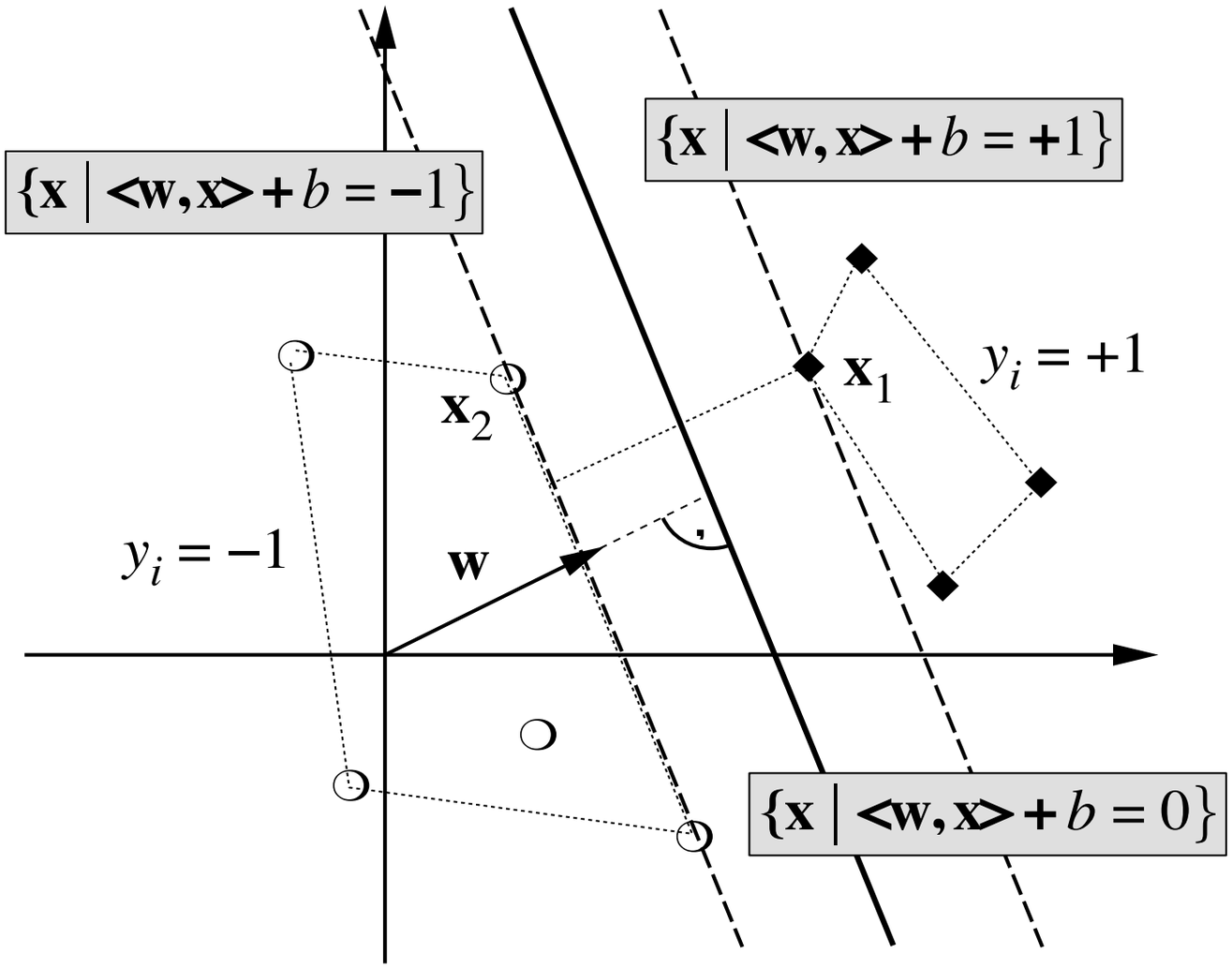
  - semidefinite constraints

$$M \succeq 0 \text{ or } M_0 + \sum_i x_i M_i \succeq 0$$

Equality is special case
Why?



Example - Support Vectors



$$\langle w, x_1 \rangle + b = 1$$
$$\langle w, x_2 \rangle + b = -1$$
$$\text{hence } \langle w, x_1 - x_2 \rangle = 2$$
$$\text{hence } \left\langle \frac{w}{\|w\|}, x_1 - x_2 \right\rangle = \frac{2}{\|w\|}$$

margin

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

Lagrange Multipliers

- Lagrange function

$$L(x, \alpha) := f(x) + \sum_{i=1}^{n} \alpha_i c_i(x) \text{ where } \alpha_i \geq 0$$

- Saddlepoint Condition

If there are x* and nonnegative α* such that

$$L(x^*, \alpha) \leq L(x^*, \alpha^*) \leq L(x, \alpha^*)$$

then x* is an optimal solution to the constrained optimization problem

Proof

$$L(x^*, \alpha) \leq L(x^*, \alpha^*) \leq L(x, \alpha^*)$$

- From first inequality we see that x* is feasible

$$(\alpha_i - \alpha_i^*) c_i(x^*) \leq 0 \text{ for all } \alpha_i \geq 0$$

- Setting some $\alpha_i = 0$ yields KKT conditions

$$\alpha_i^* c_i(x^*) = 0$$

- Consequently we have

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

This proves optimality

Constraint gymnastics (all three conditions are equivalent)

- **Slater's condition**
  There exists some x such that for all i

$$c_i(x) < 0$$

- **Karlin's condition**
  For all nonnegative α there exists some x such that

$$\sum_i \alpha_i c_i(x) \leq 0$$

- **Strict constraint qualification**
  The feasible region contains at least two distinct elements and there exists an x in X such that all $c_i(x)$ are strictly convex at x with respect to X

Necessary Kuhn-Tucker Conditions

- Assume optimization problem
  - satisfies the constraint qualifications
  - has convex differentiable objective + constraints
- Then the KKT conditions are necessary & sufficient

$$\partial_x L(x^*, \alpha^*) = \partial_x f(x^*) + \sum_i \alpha_i^* \partial_x c_i(x^*) \quad = 0 \text{ (Saddlepoint in } x^*)$$

$$\partial_{\alpha_i} L(x^*, \alpha^*) = c_i(x^*) \quad \leq 0 \text{ (Saddlepoint in } \alpha^*)$$

$$\sum_i \alpha_i^* c_i(x^*) \quad = 0 \text{ (Vanishing KKT-gap)}$$

Yields algorithm for solving optimization problems
Solve for saddlepoint and KKT conditions

Proof

$$
\begin{aligned}
f(x) - f(x^*) &\geq \left[\partial_x f(x^*)\right]^\top (x - x^*) && \text{(by convexity)} \\
&= -\sum_i \alpha_i^* \left[\partial_x c_i(x^*)\right]^\top (x - x^*) && \text{(by Saddlepoint in } x^*\text{)} \\
&\geq -\sum_i \alpha_i^* (c_i(x) - c_i(x^*)) && \text{(by convexity)} \\
&= \sum_i \alpha_i^* c_i(x) && \text{(by vanishing KKT gap)} \\
&\geq 0
\end{aligned}
$$

Linear and Quadratic Programs

Linear Programs

- Objective

$$\underset{x}{\text{minimize}}\, c^\top x \text{ subject to } Ax + d \leq 0$$

- Lagrange function

$$L(x, \alpha) = c^\top x + \alpha^\top (Ax + d)$$

- Optimality conditions

$$\partial_x L(x, \alpha) = A^\top \alpha + c = 0$$
$$\partial_\alpha L(x, \alpha) = Ax + d \leq 0$$
$$0 = \alpha^\top (Ax + d)$$
$$0 \leq \alpha$$

- Dual problem

$$\underset{i}{\text{maximize}}\, d^\top \alpha \text{ subject to } A^\top \alpha + c = 0 \text{ and } \alpha \geq 0$$

Linear Programs

- Objective

$$\underset{x}{\text{minimize}}\, c^\top x \text{ subject to } Ax + d \leq 0$$

- Lagrange function

$$L(x, \alpha) = c^\top x + \alpha^\top (Ax + d)$$

- Optimality conditions

$$\partial_x L(x, \alpha) = A^\top \alpha + c = 0$$
$$\partial_\alpha L(x, \alpha) = Ax + d \leq 0$$
$$0 = \alpha^\top (Ax + d)$$
$$0 \leq \alpha$$

plug into L

- Dual problem

$$\underset{i}{\text{maximize}}\, d^\top \alpha \text{ subject to } A^\top \alpha + c = 0 \text{ and } \alpha \geq 0$$

Linear Programs

- Objective

$$\underset{x}{\text{minimize}}\, c^\top x \text{ subject to } Ax + d \leq 0$$

- Lagrange function

$$L(x, \alpha) = c^\top x + \alpha^\top (Ax + d)$$

- Optimality conditions

$$\partial_x L(x, \alpha) = A^\top \alpha + c = 0$$
$$\partial_\alpha L(x, \alpha) = Ax + d \leq 0$$
$$0 = \alpha^\top (Ax + d)$$
$$0 \leq \alpha$$

plug into L

- Dual problem

$$\underset{i}{\text{maximize}}\, d^\top \alpha \text{ subject to } A^\top \alpha + c = 0 \text{ and } \alpha \geq 0$$

Linear Programs

- Primal

$$\underset{x}{\text{minimize}}\, c^\top x \text{ subject to } Ax + d \leq 0$$

- Dual

$$\underset{i}{\text{maximize}}\, d^\top \alpha \text{ subject to } A^\top \alpha + c = 0 \text{ and } \alpha \geq 0$$

- Free variables become equality constraints
- Equality constraints become free variables
- Inequalities become inequalities
- Dual of dual is primal

Quadratic Programs

- Objective

$$\underset{x}{\text{minimize}}\ \frac{1}{2}x^\top Qx + c^\top x \text{ subject to } Ax + d \leq 0$$

- Lagrange function

$$L(x, \alpha) = \frac{1}{2}x^\top Qx + c^\top x + \alpha^\top (Ax + d)$$

- Optimality conditions

$$\partial_x L(x, \alpha) = Qx + A^\top \alpha + c = 0$$
$$\partial_\alpha L(x, \alpha) = Ax + d \leq 0$$
$$0 = \alpha^\top (Ax + d)$$
$$0 \leq \alpha$$

plug into L

Quadratic Program

- Eliminating x from the Lagrangian via

$$Qx + A^\top \alpha + c = 0$$

- Lagrange function

$$\begin{aligned} L(x, \alpha) &= \frac{1}{2} x^\top Q x + c^\top x + \alpha^\top (Ax + d) \\ &= -\frac{1}{2} x^\top Q x + \alpha^\top d \\ &= -\frac{1}{2} (A^\top \alpha + c)^\top Q^{-1} (A^\top \alpha + c) + \alpha^\top d \\ &= -\frac{1}{2} \alpha^\top A Q^{-1} A^\top \alpha + \alpha^\top \left[d - A Q^{-1} c \right] - \frac{1}{2} c^\top Q^{-1} c \end{aligned}$$

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

Quadratic Program

- Eliminating x from the Lagrangian via

$$Qx + A^\top \alpha + c = 0$$

- Lagrange function

$$\begin{aligned} L(x,\alpha) &= \frac{1}{2} x^\top Q x + c^\top x + \alpha^\top (Ax + d) \\ &= -\frac{1}{2} x^\top Q x + \alpha^\top d \\ &= -\frac{1}{2} (A^\top \alpha + c)^\top Q^{-1} (A^\top \alpha + c) + \alpha^\top d \\ &= -\frac{1}{2} \alpha^\top A Q^{-1} A^\top \alpha + \alpha^\top \left[d - A Q^{-1} c \right] - \frac{1}{2} c^\top Q^{-1} c \end{aligned}$$

dual

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

Quadratic Programs

- Primal

$$\underset{x}{\text{minimize}}\ \frac{1}{2}x^\top Qx + c^\top x \text{ subject to } Ax + d \leq 0$$

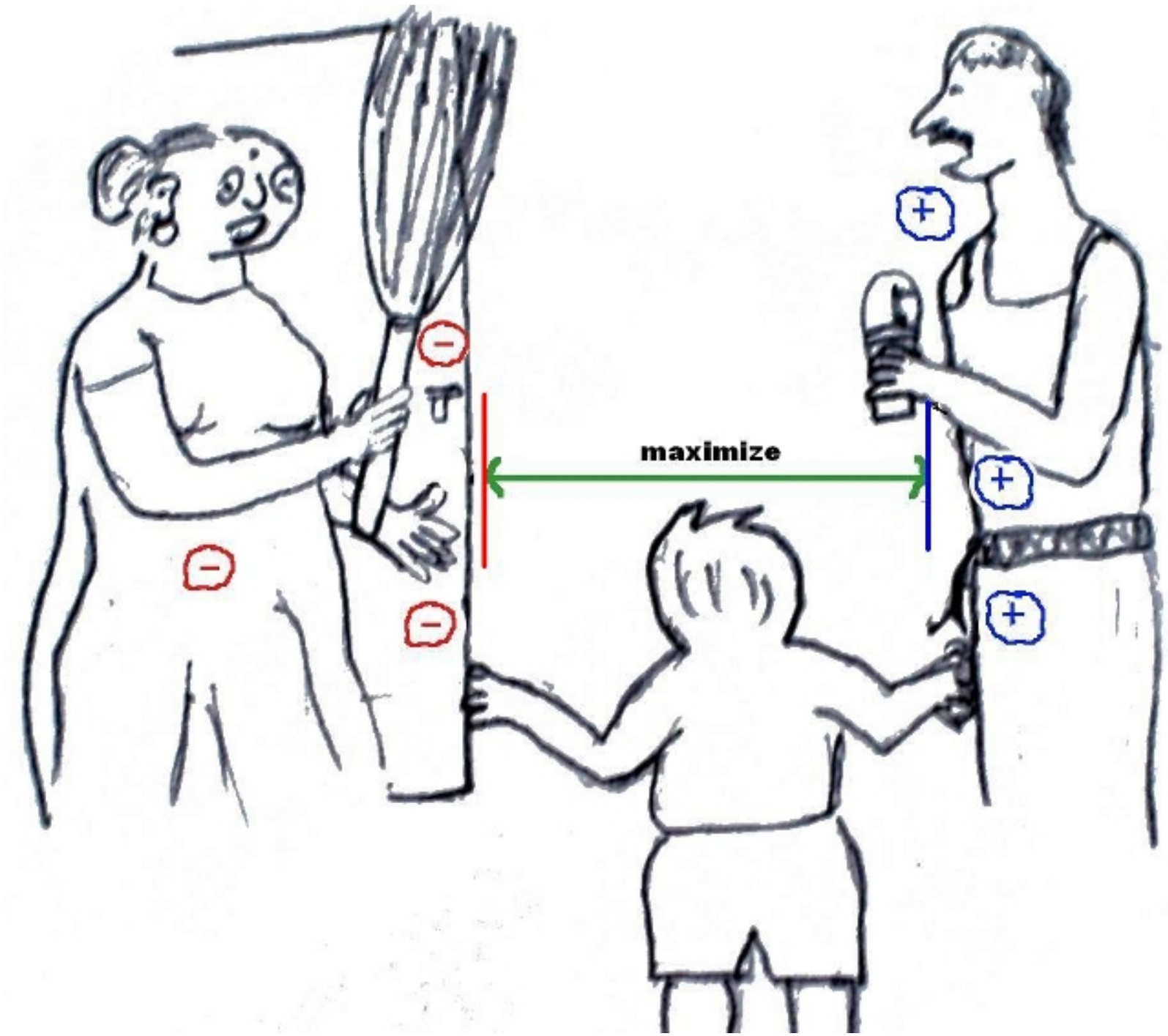
- Dual

$$\underset{\alpha}{\text{minimize}}\ \frac{1}{2}\alpha^\top AQ^{-1}A^\top\alpha + \alpha^\top\left[AQ^{-1}c - d\right] \text{ subject to } \alpha \geq 0$$

- Dual constraints are simpler
- Possibly many fewer variables
- Dual of dual is not (always) primal
  (e.g. in SVMs x is in a Hilbert Space)

Outline

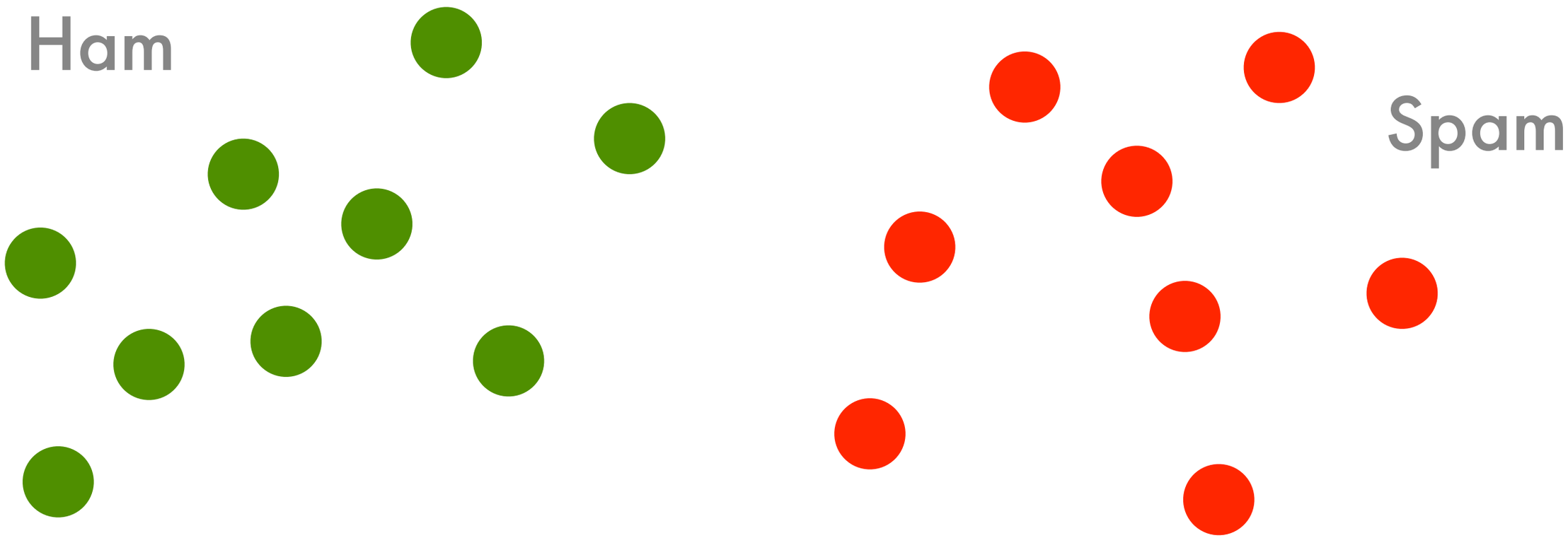
- **Convex Optimization**
  - Unconstrained Optimization
  - Constrained Optimization and Duality
  - Linear and Quadratic programs
- **Support Vector Machines**
  - Classification
  - Regression
  - Novelty Detection
- **Kernels**
  - Feature Space
  - Kernel PCA
  - Kernelized SVM



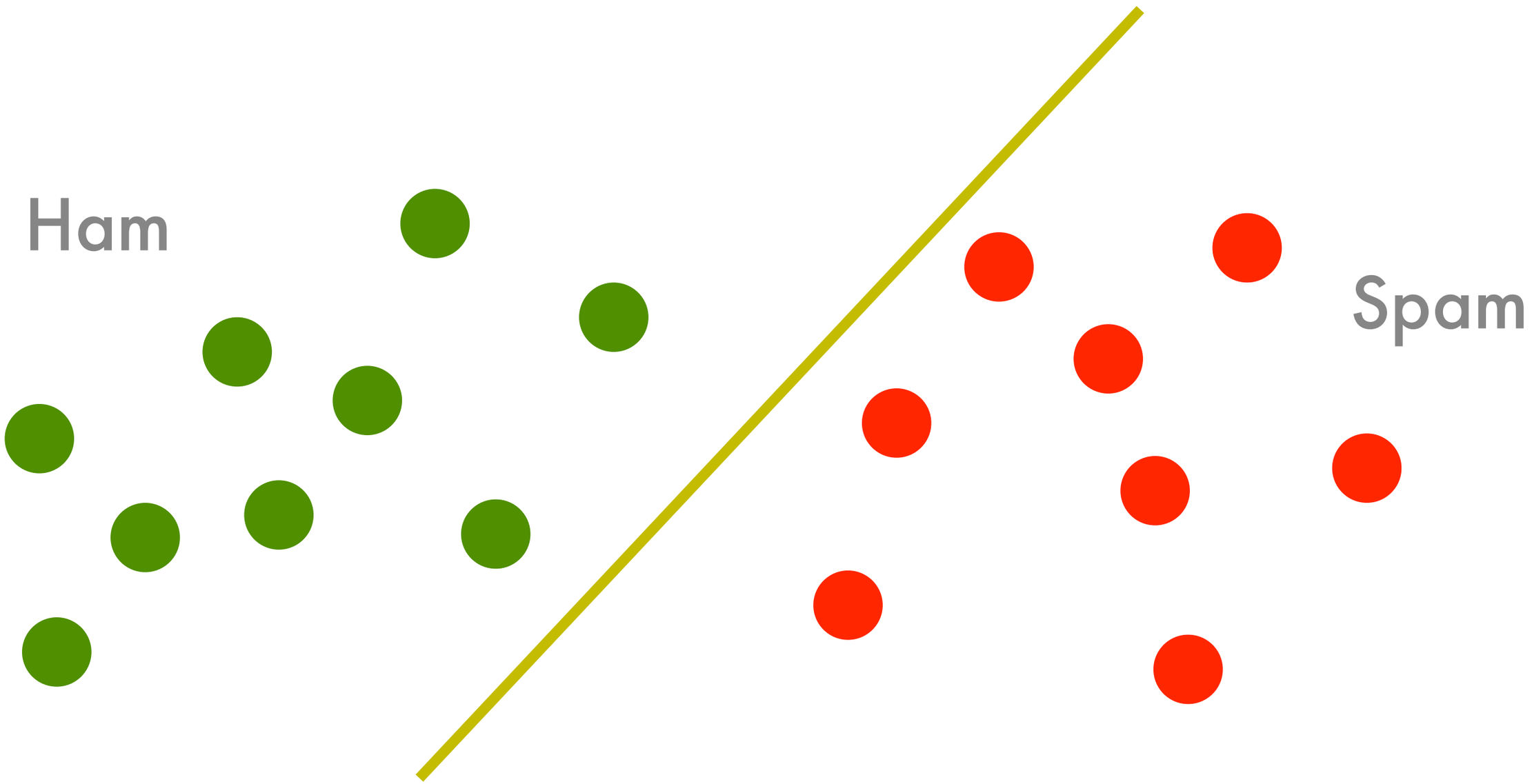
Support Vector Machines

Carnegie Mellon University

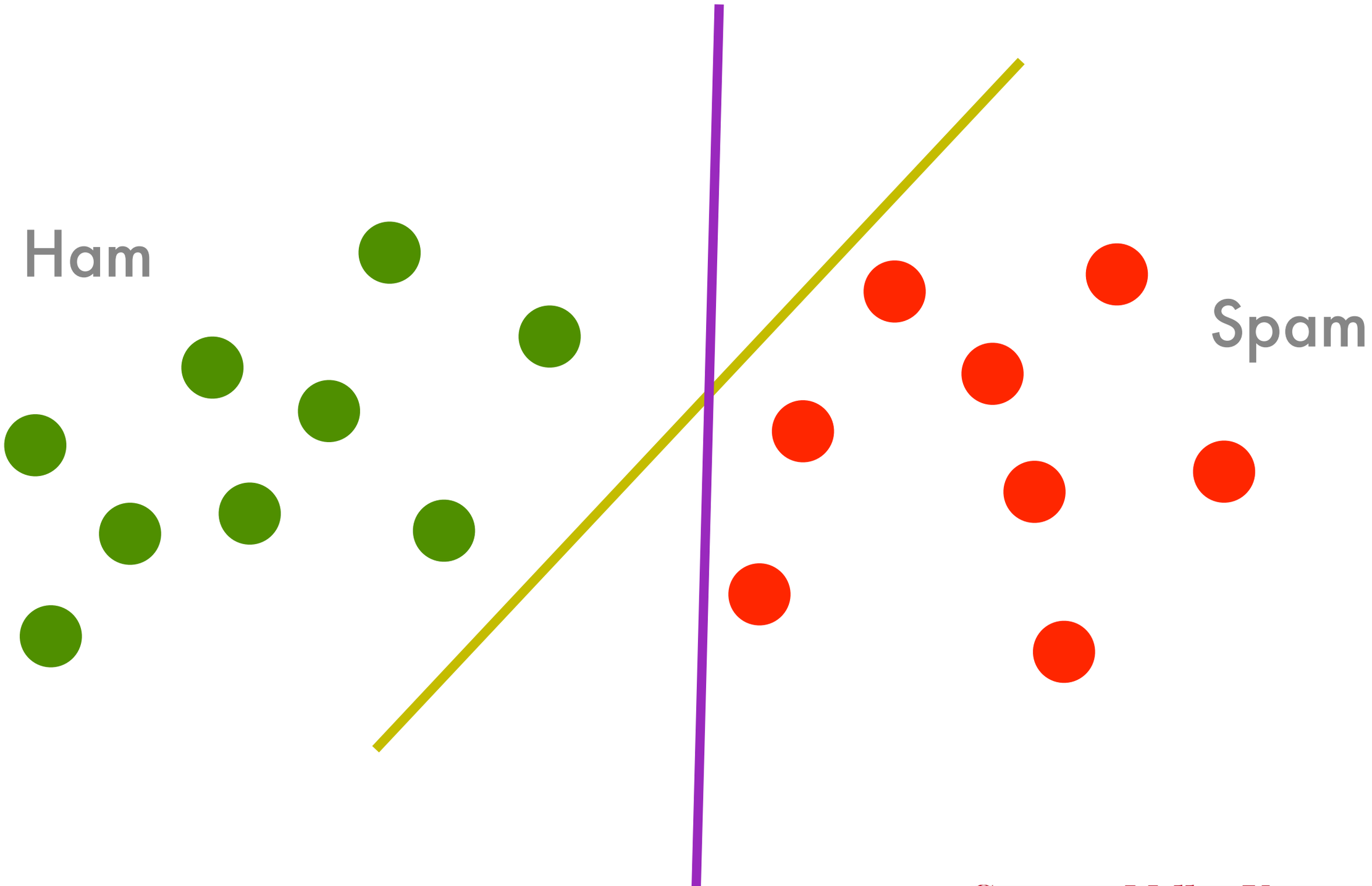
Linear Separator



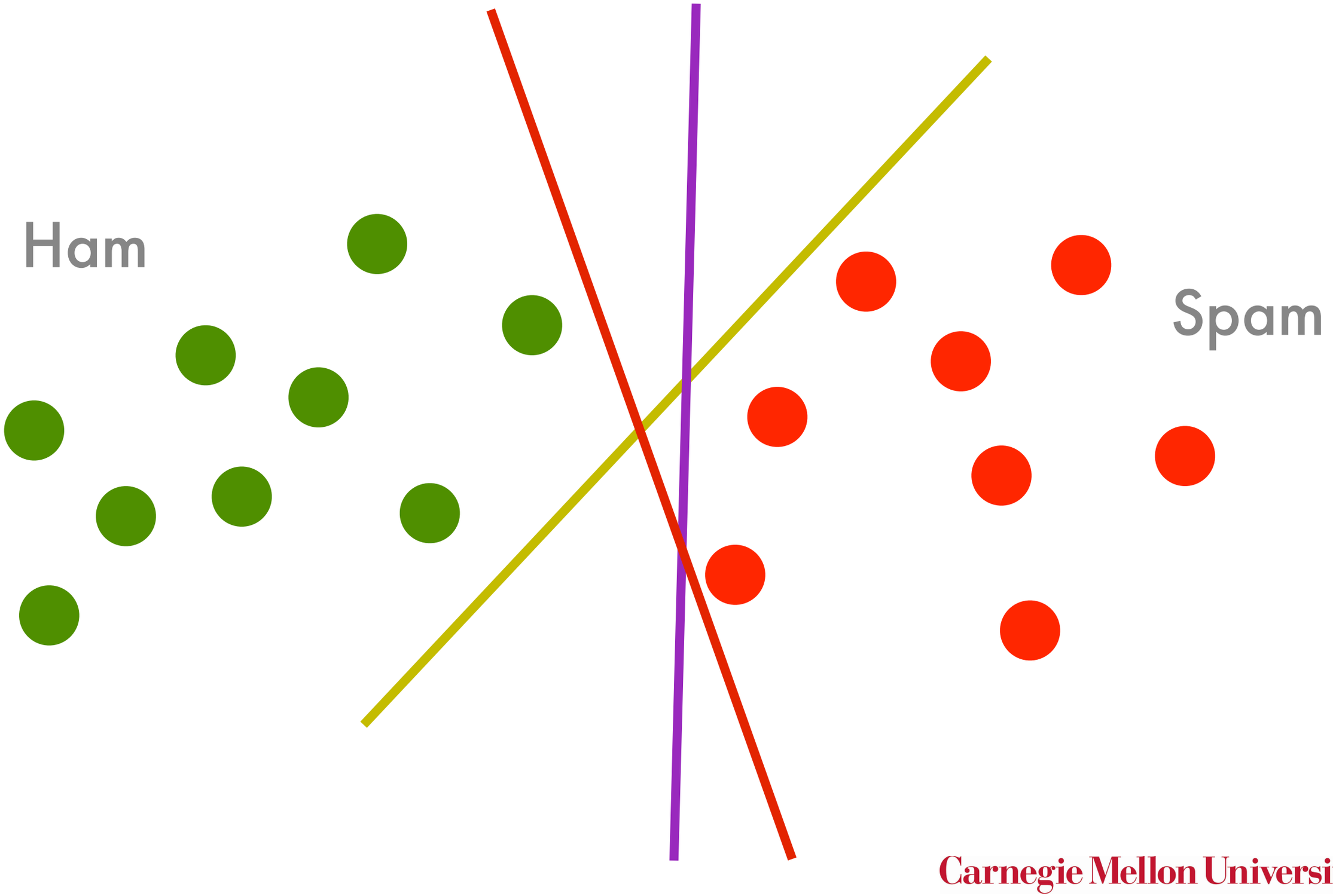
Linear Separator



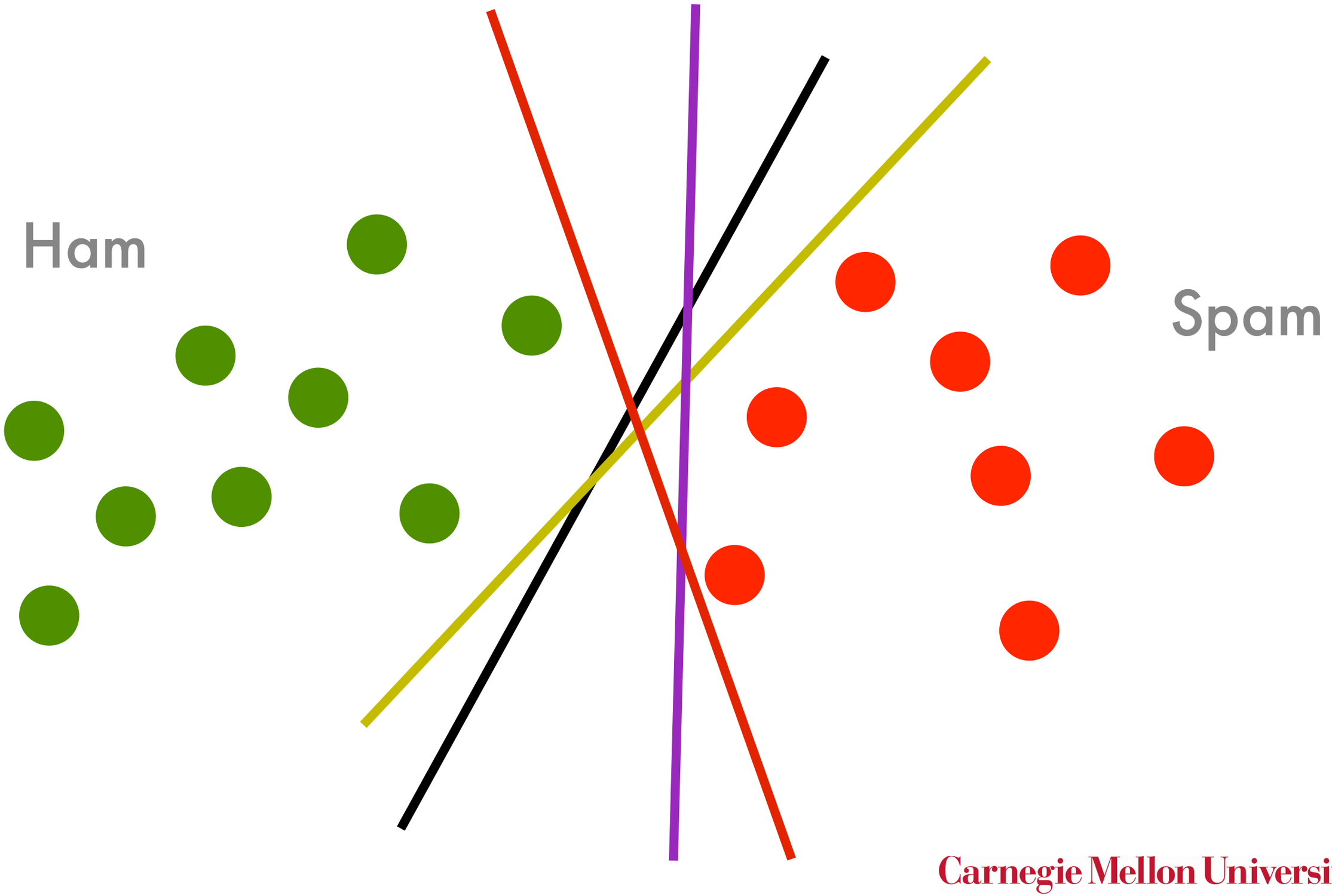
Linear Separator



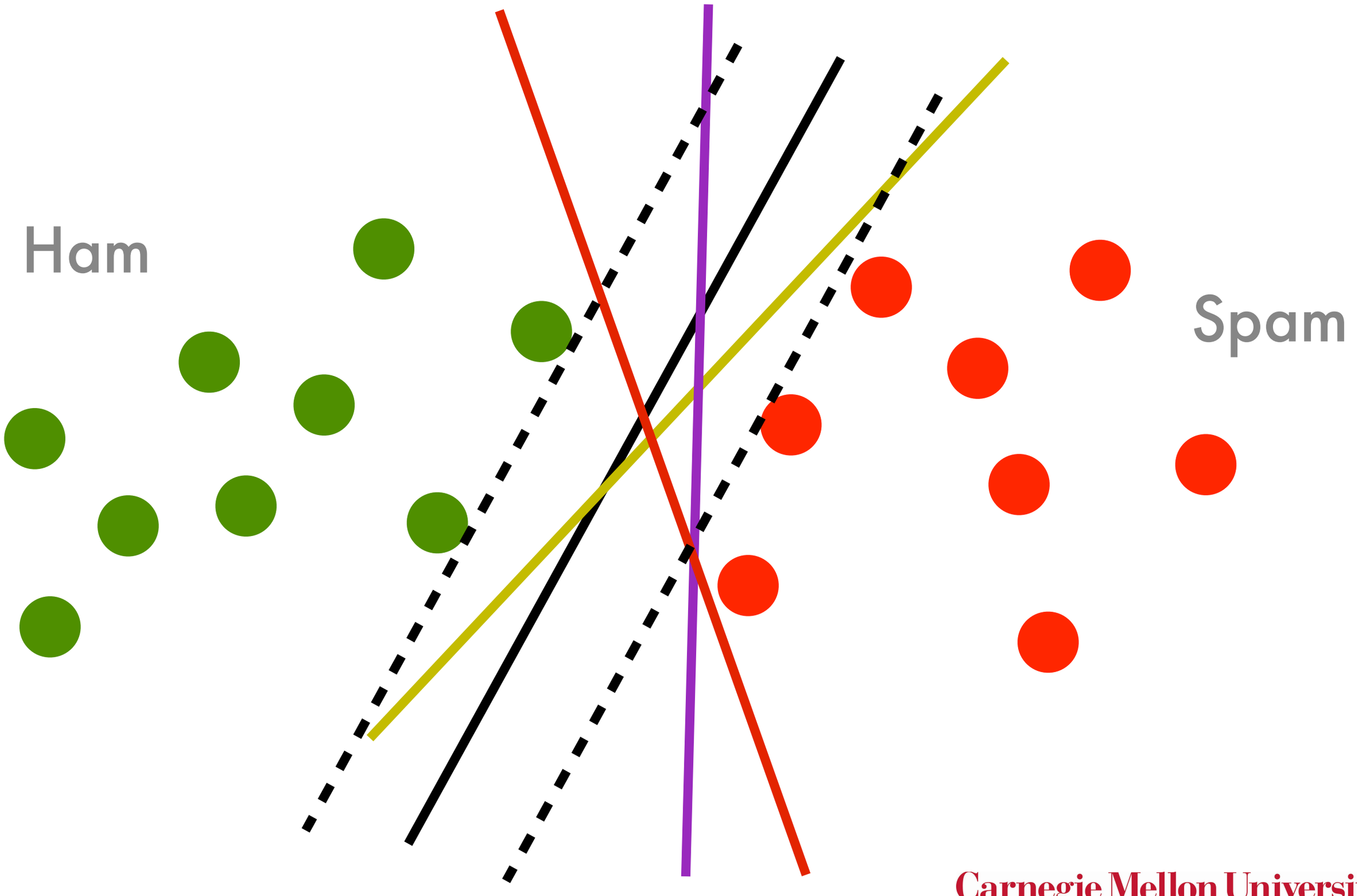
Linear Separator



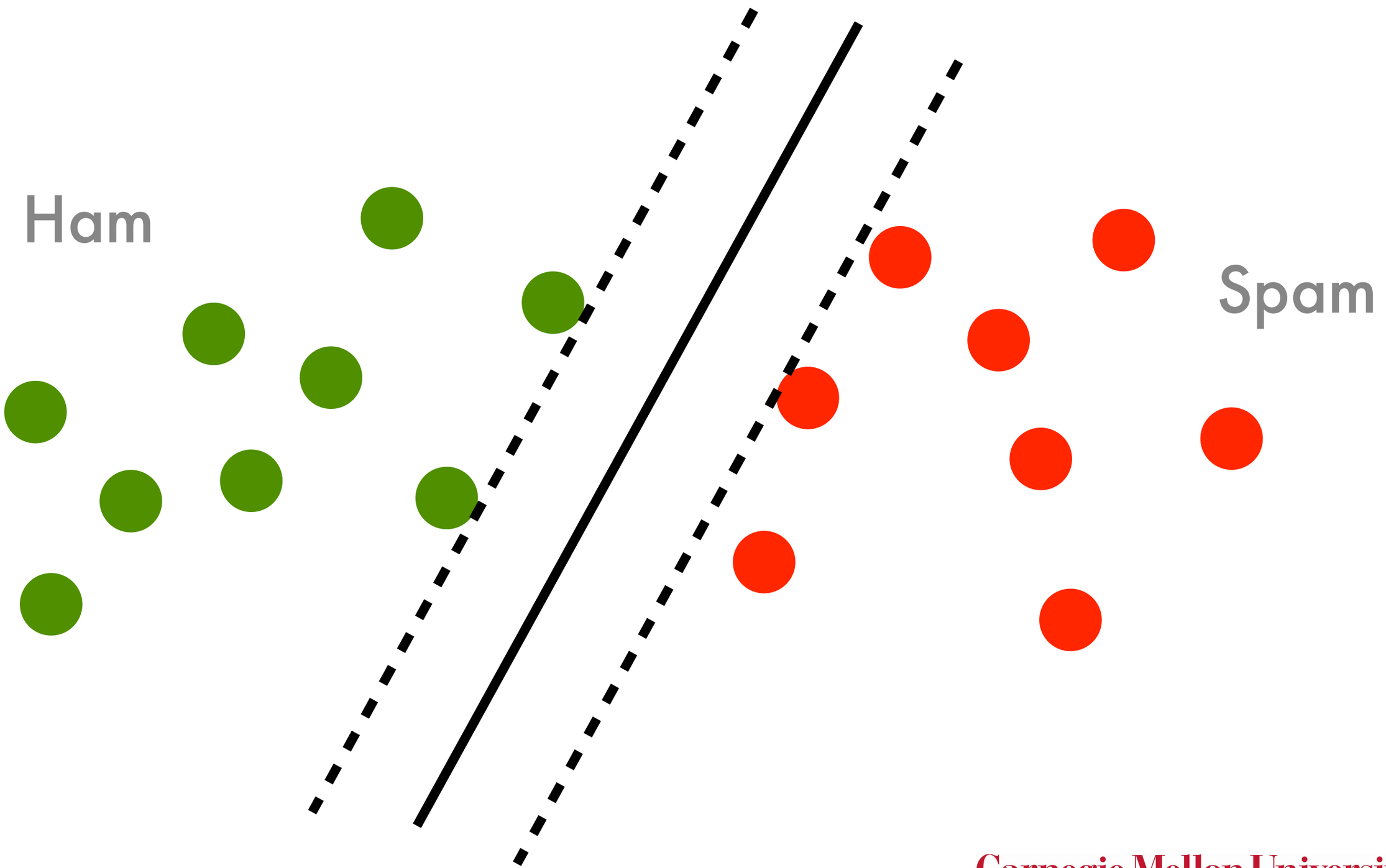
Linear Separator



Linear Separator



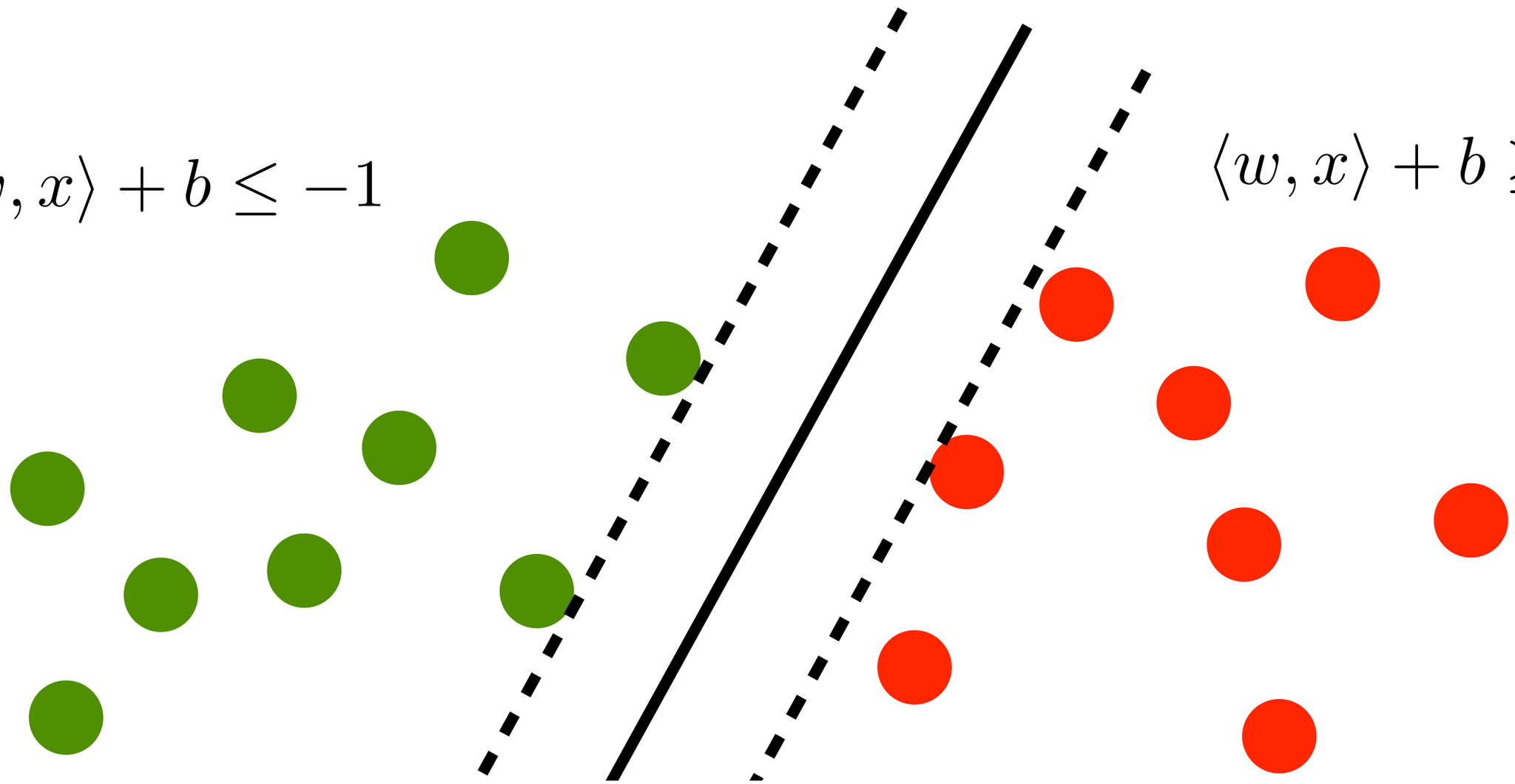
Linear Separator



Large Margin Classifier

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

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



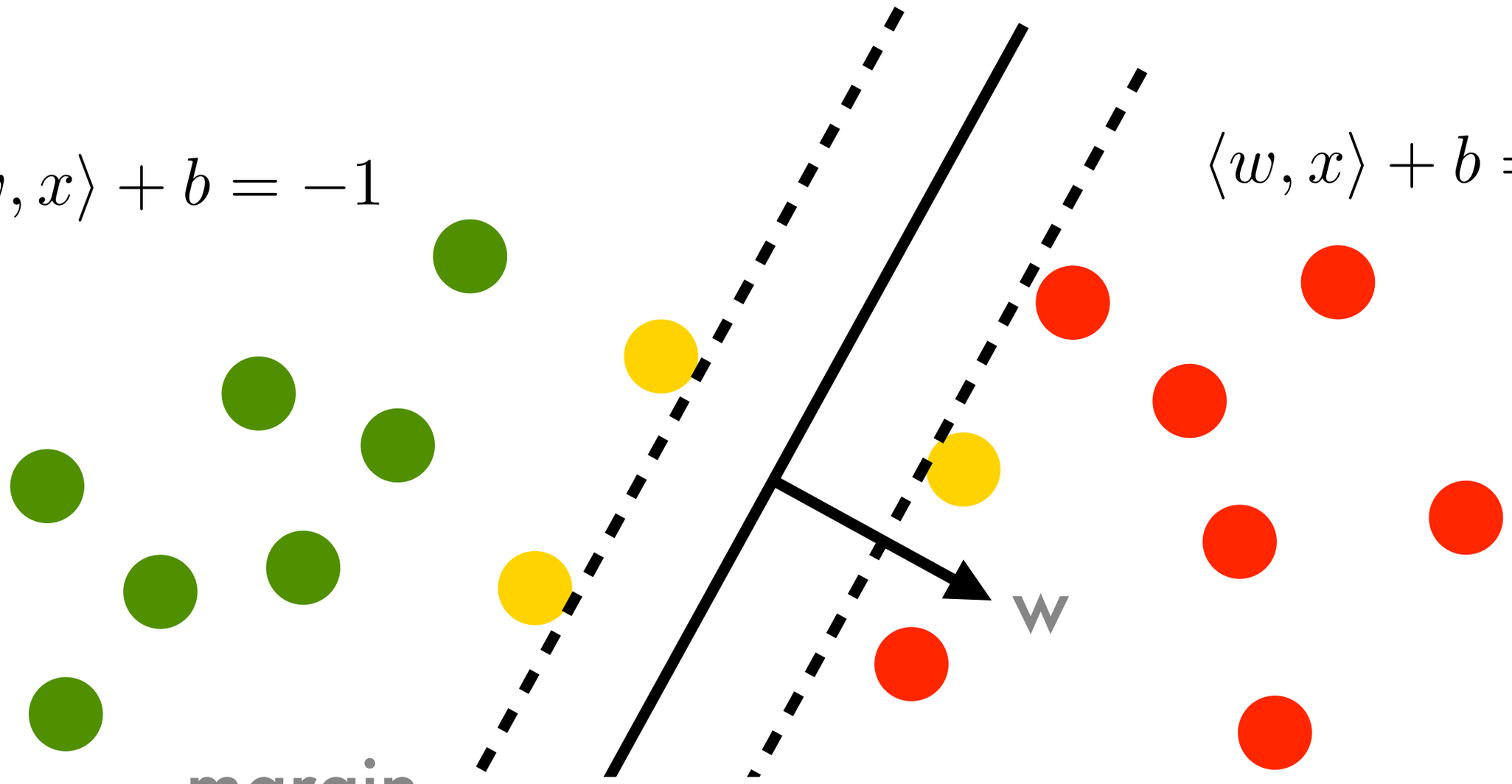
linear function

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

Large Margin Classifier

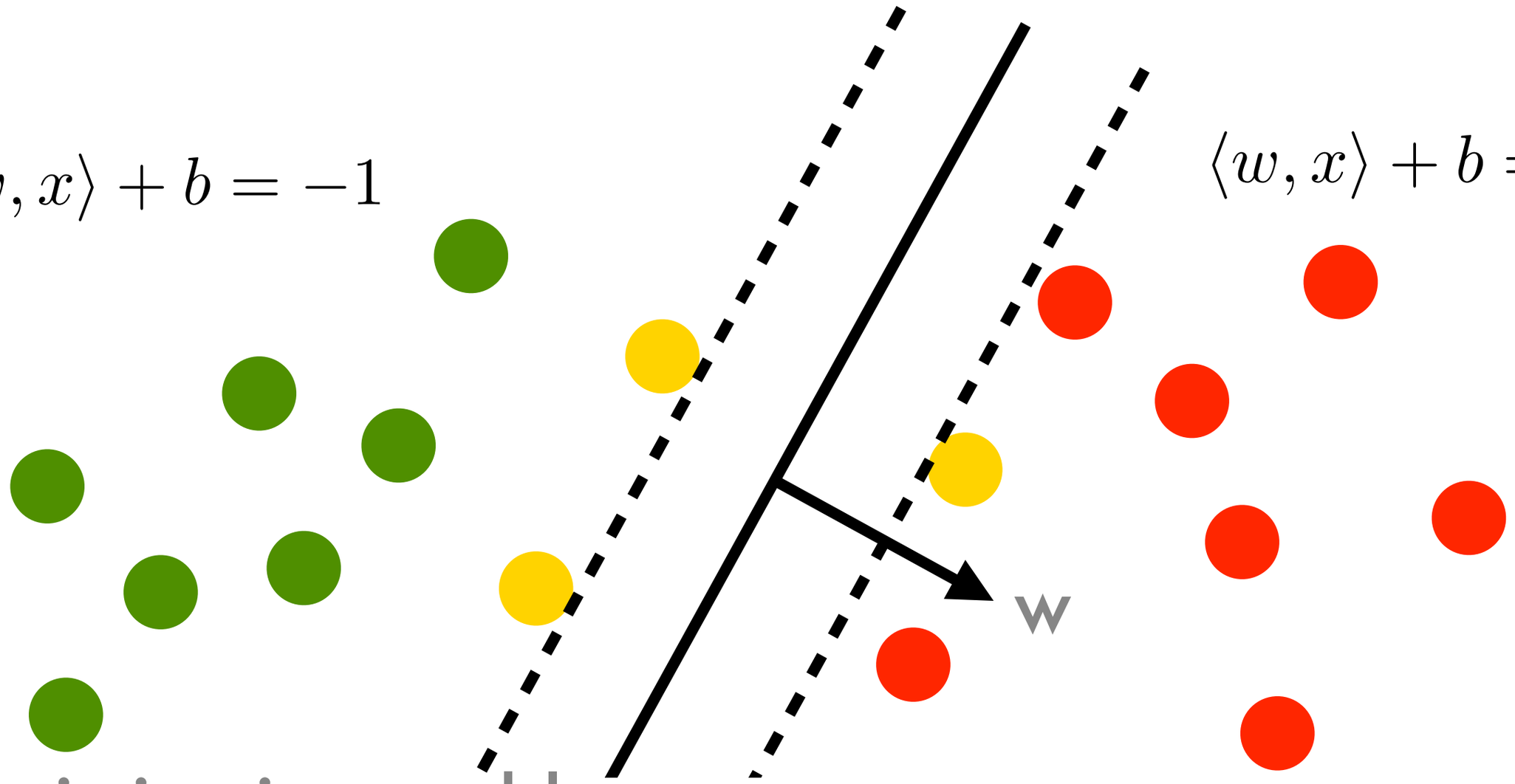
$\langle w, x \rangle + b = -1$

$\langle w, x \rangle + b = 1$



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

Large Margin Classifier



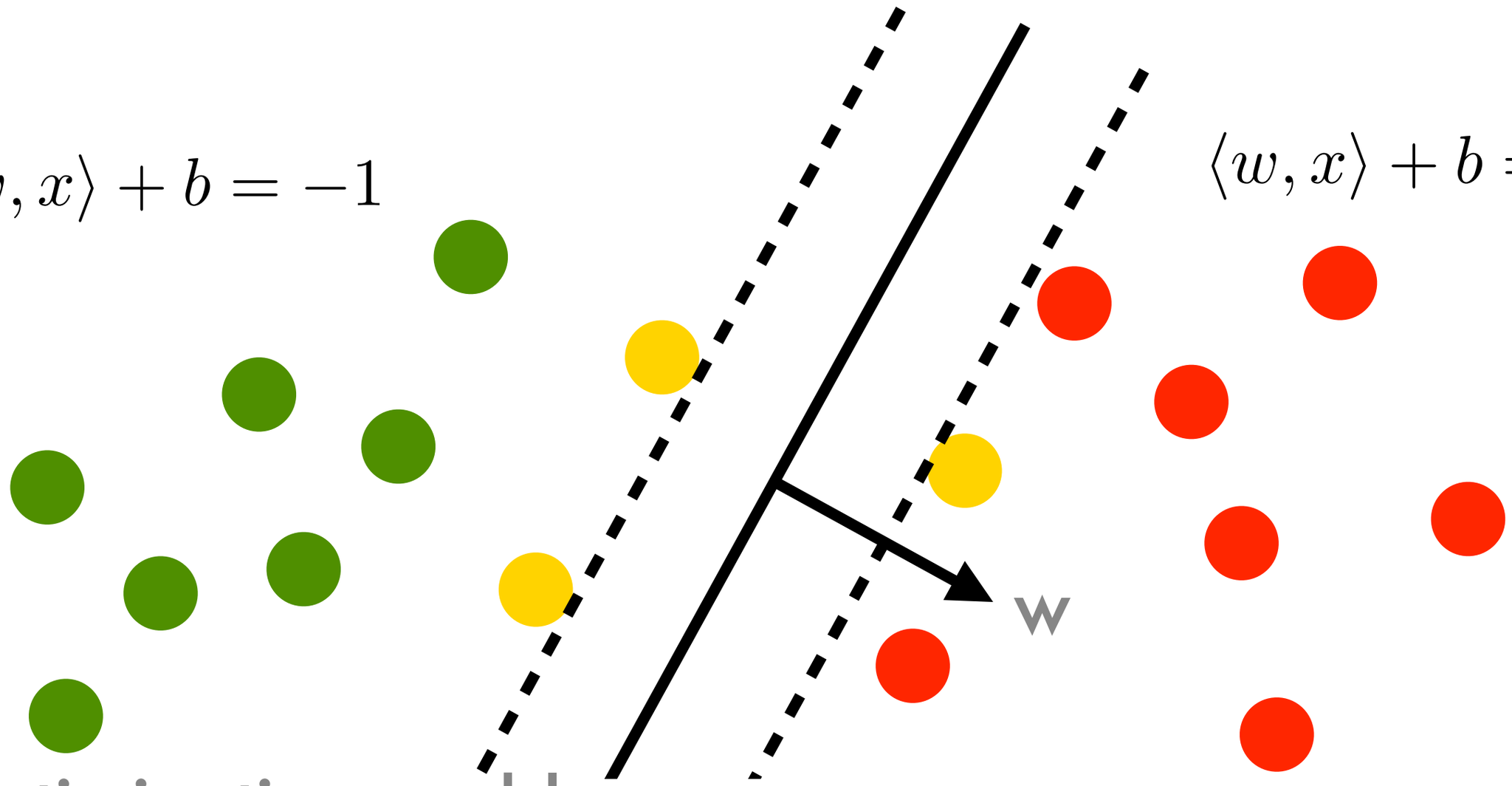
optimization problem

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

Large Margin Classifier

$\langle w, x \rangle + b = -1$

$\langle w, x \rangle + b = 1$



optimization problem

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

Dual Problem

- Primal optimization problem

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

- Lagrange function

constraint

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

Optimality in w, b is at saddle point with α

- Derivatives in w, b need to vanish

Dual Problem

- Lagrange function

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

- Derivatives in w, b need to vanish

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

$$\partial_b L(w, b, a) = \sum_i \alpha_i y_i = 0$$

- Plugging terms back into L yields

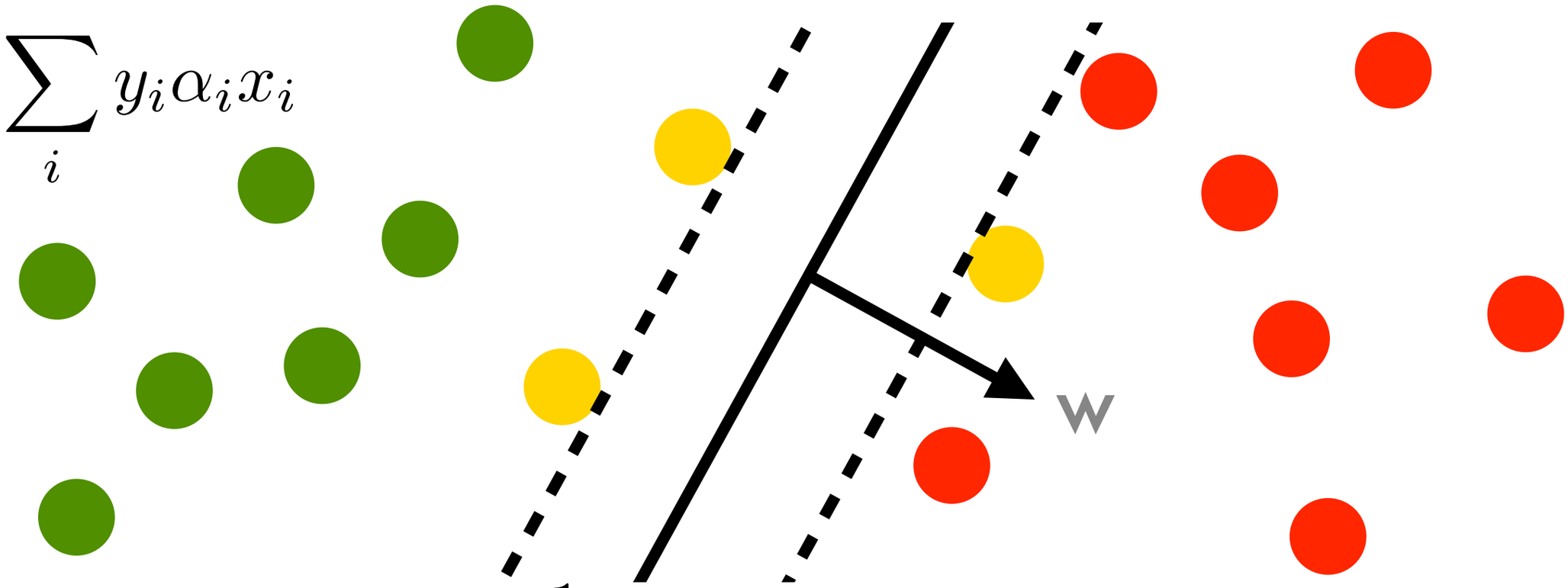
$$\underset{\alpha}{\text{maximize}} -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i$$

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

Support Vector Machines

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

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



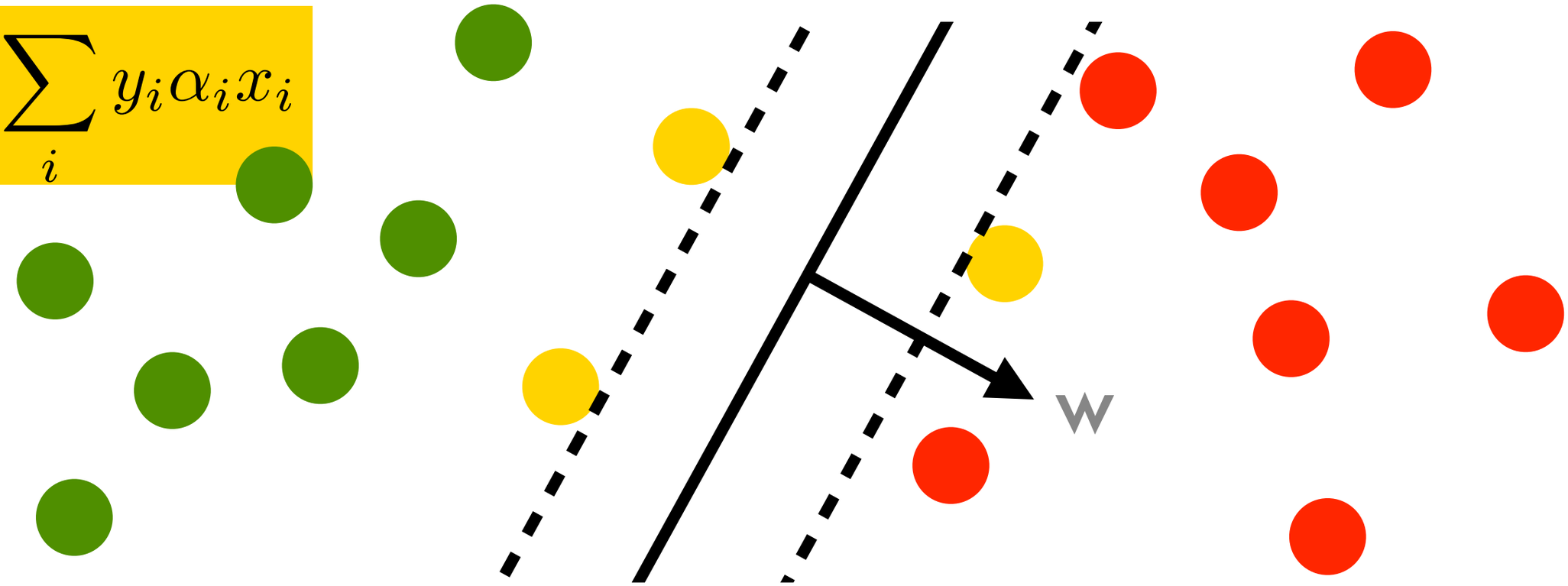
$$\underset{\alpha}{\text{maximize}} -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j\rangle + \sum_i \alpha_i$$

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

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 Optimality condition

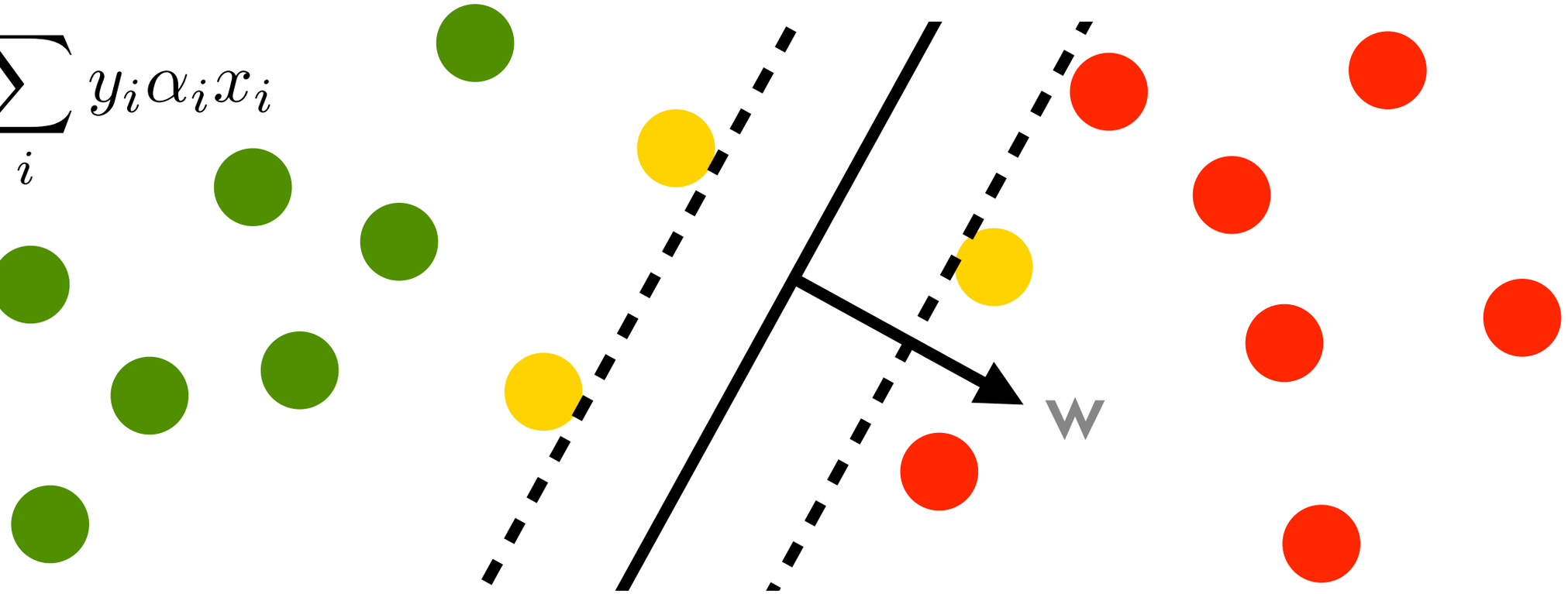
$$\alpha_i \left[y_i \left[\langle w, x_i\rangle + b\right] - 1\right] = 0$$

$$\alpha_i = 0$$

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

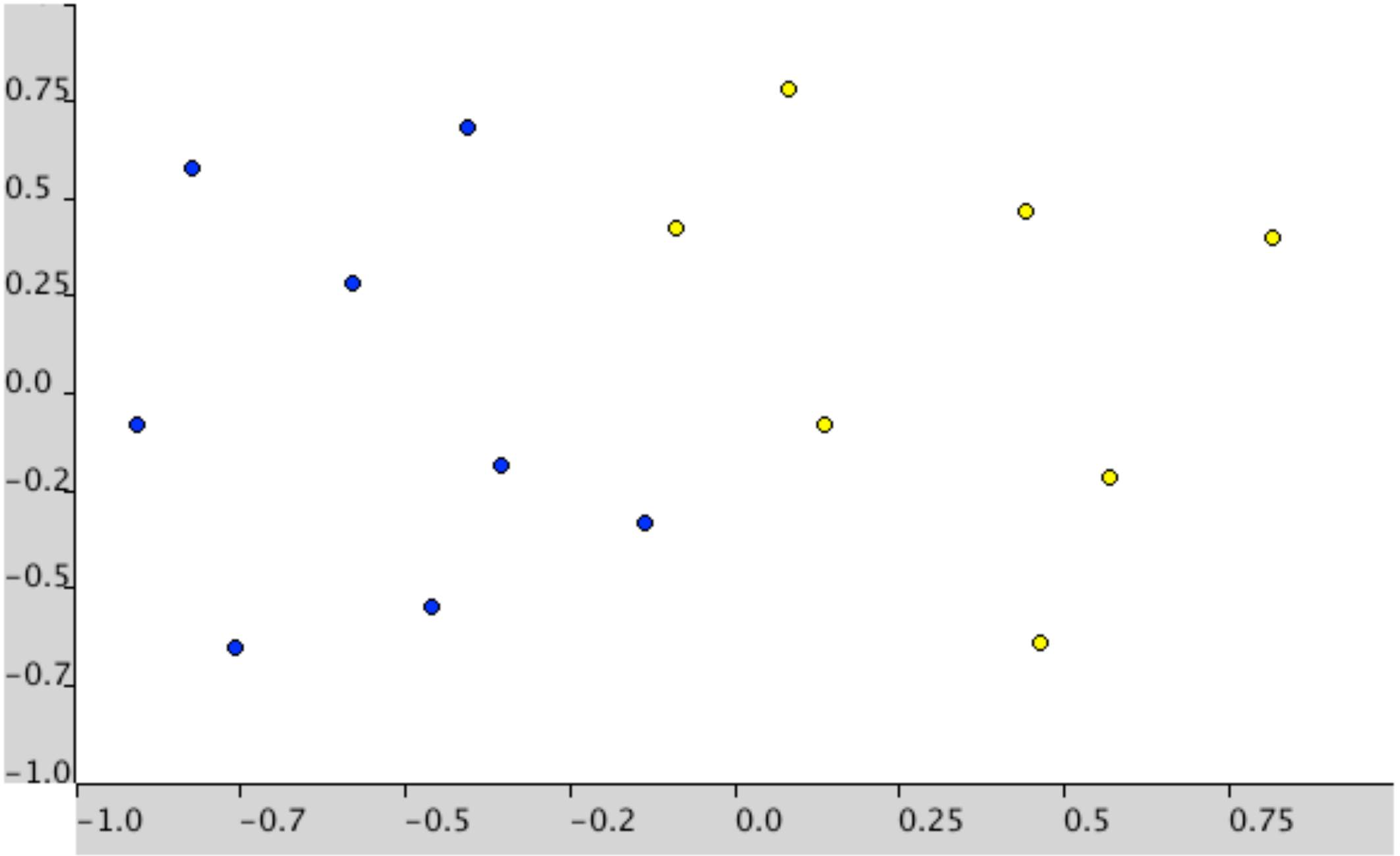
Properties

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



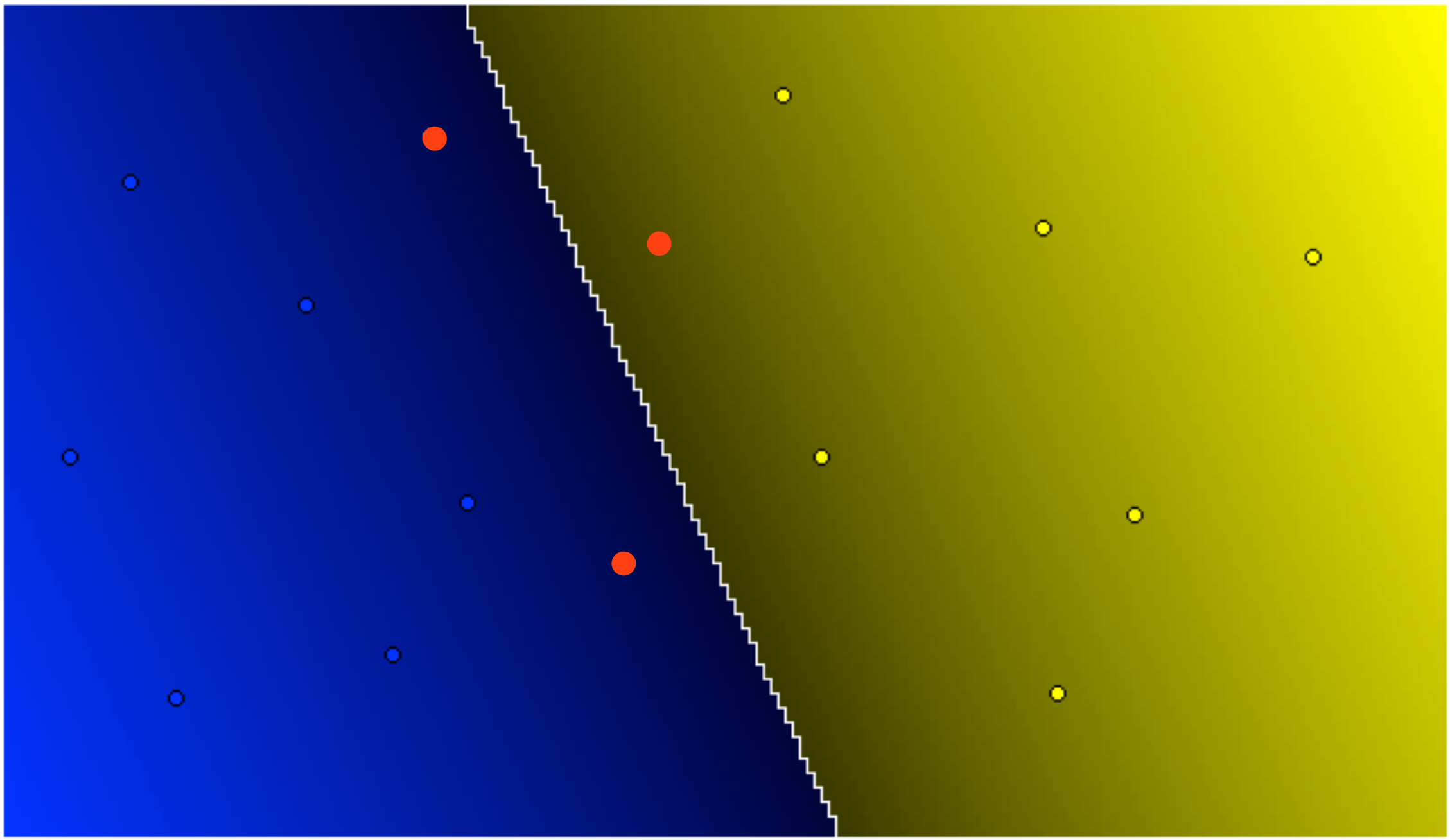
- Weight vector w as weighted linear combination of instances
- Only points on margin matter (ignore the rest and get same solution)
- Only inner products matter
  - Quadratic program
  - We can replace the inner product by a kernel
- Keeps instances away from the margin

Example

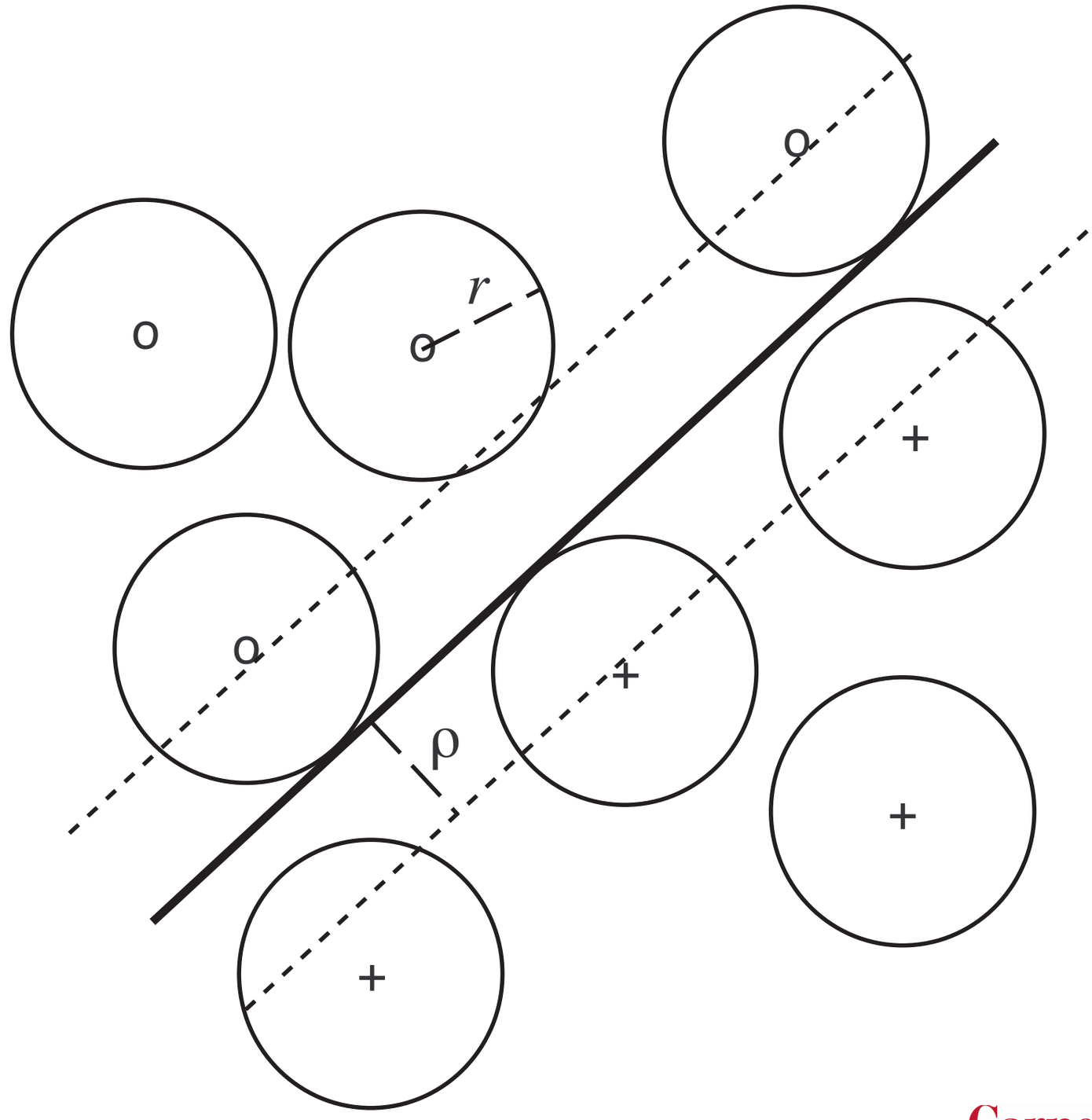


Example

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



Why large margins?

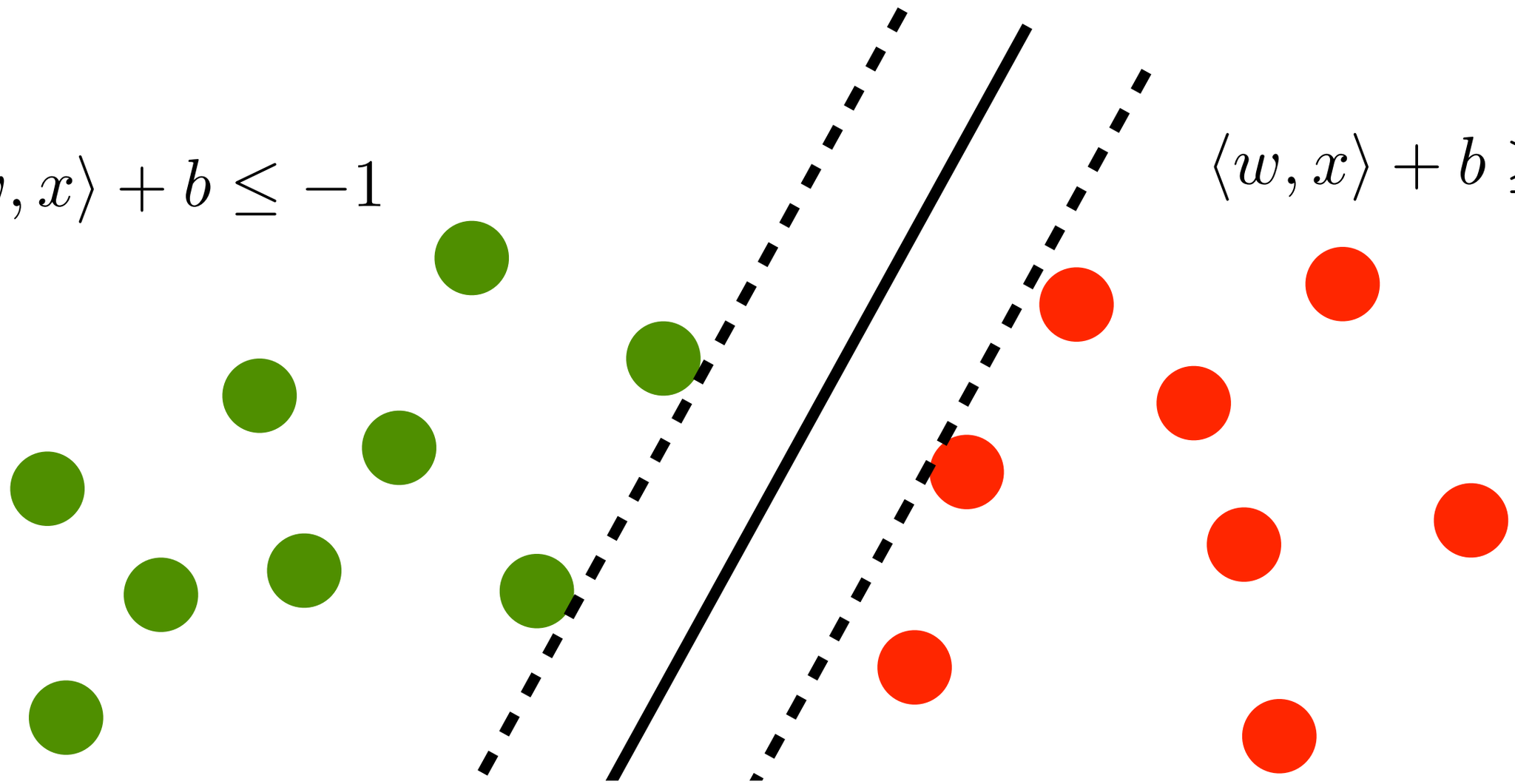



SOFT
MARC
CLASSIFIE
ERS

Large Margin Classifier

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

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



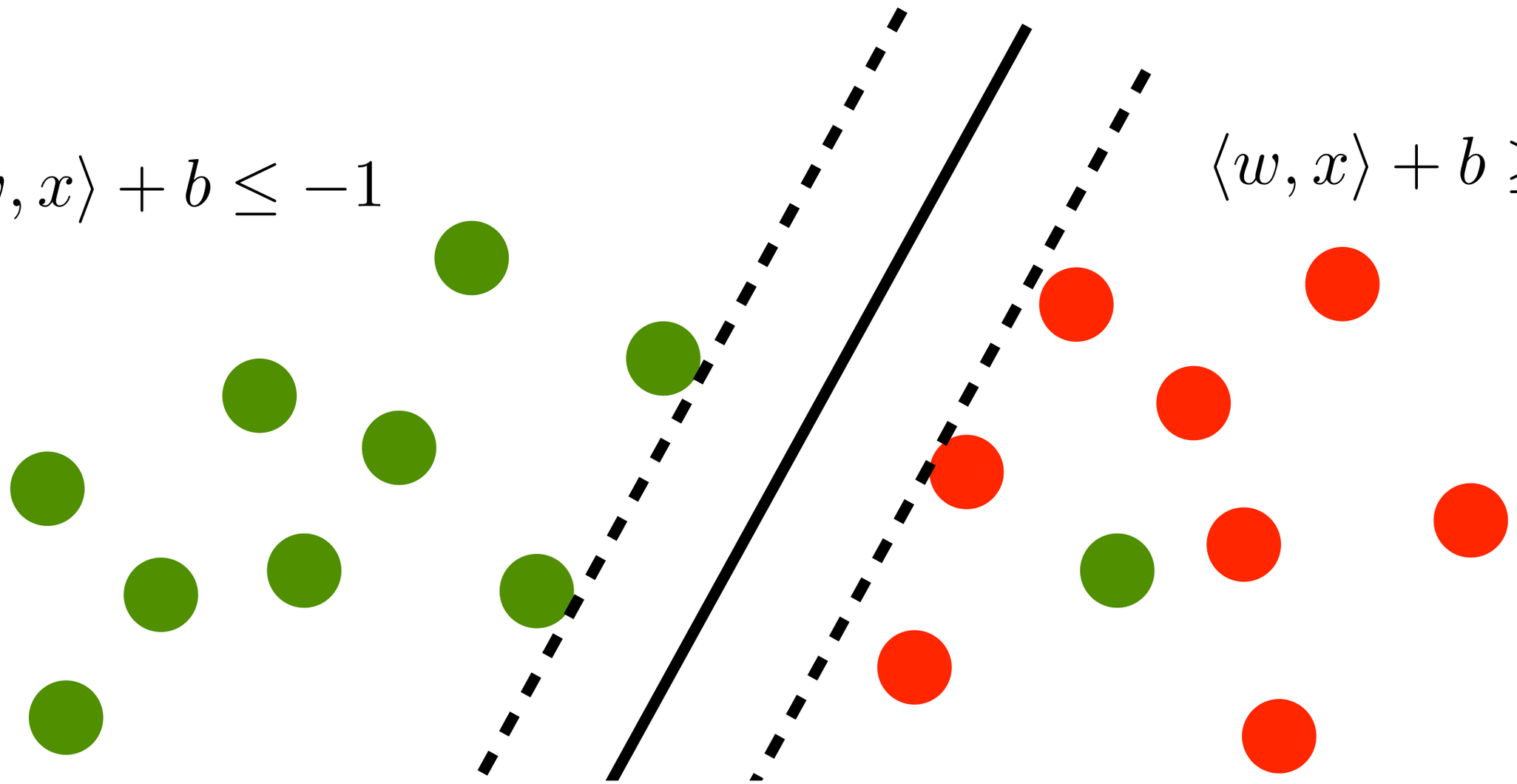
linear function

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

Large Margin Classifier

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

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



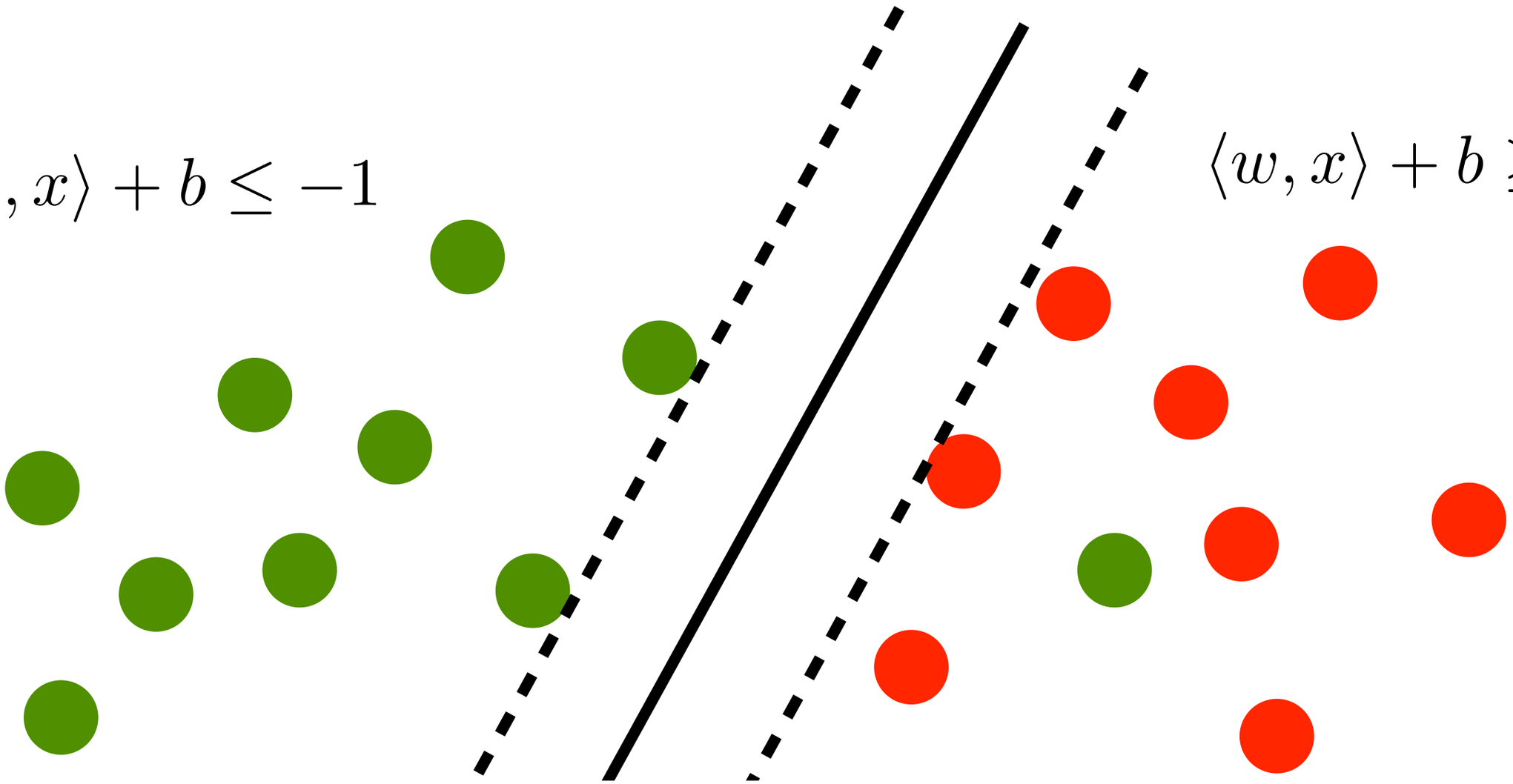
linear function

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

Large Margin Classifier

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

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



linear function

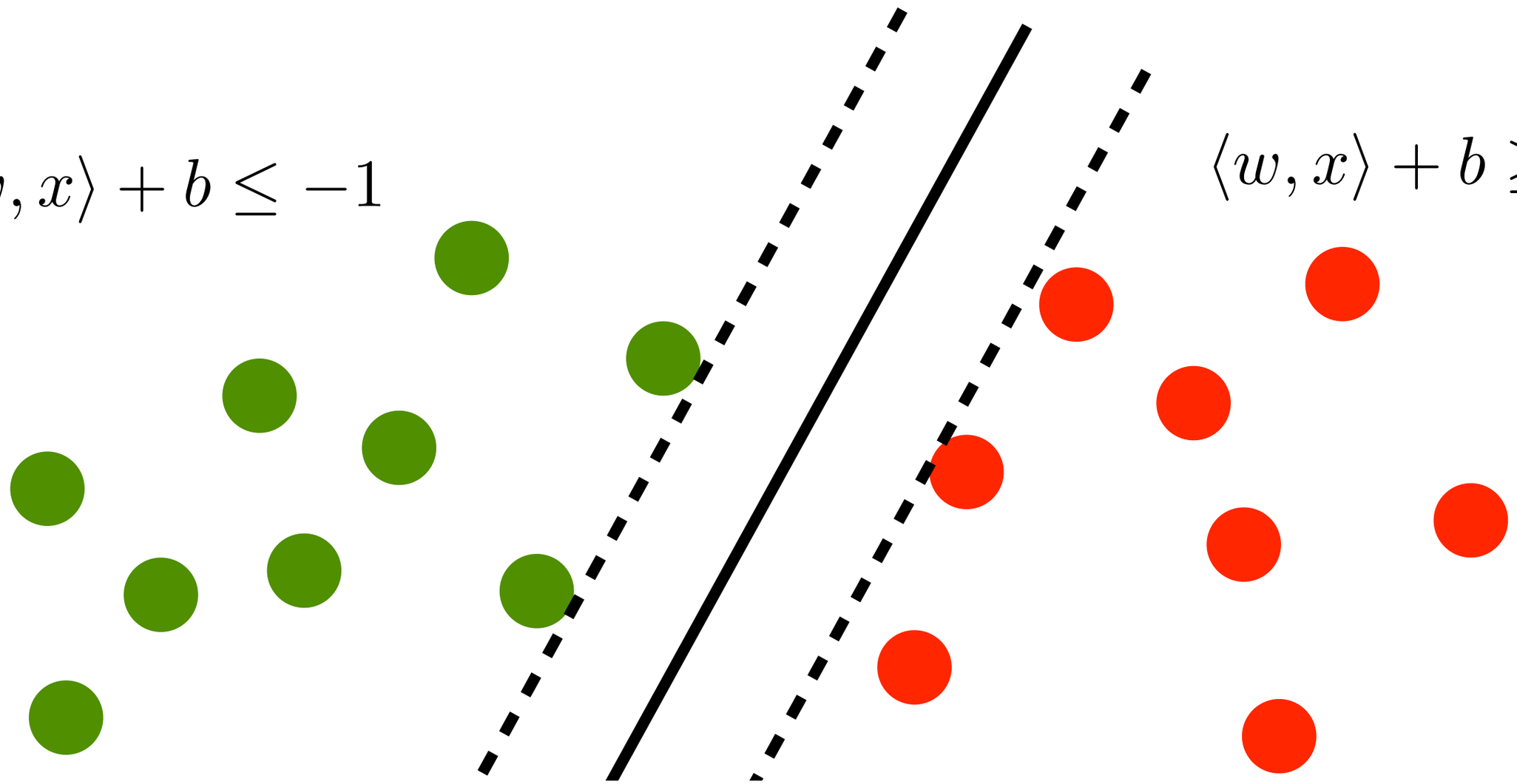
$f(x) = \langle w, x \rangle + b$

linear separator is impossible

Large Margin Classifier

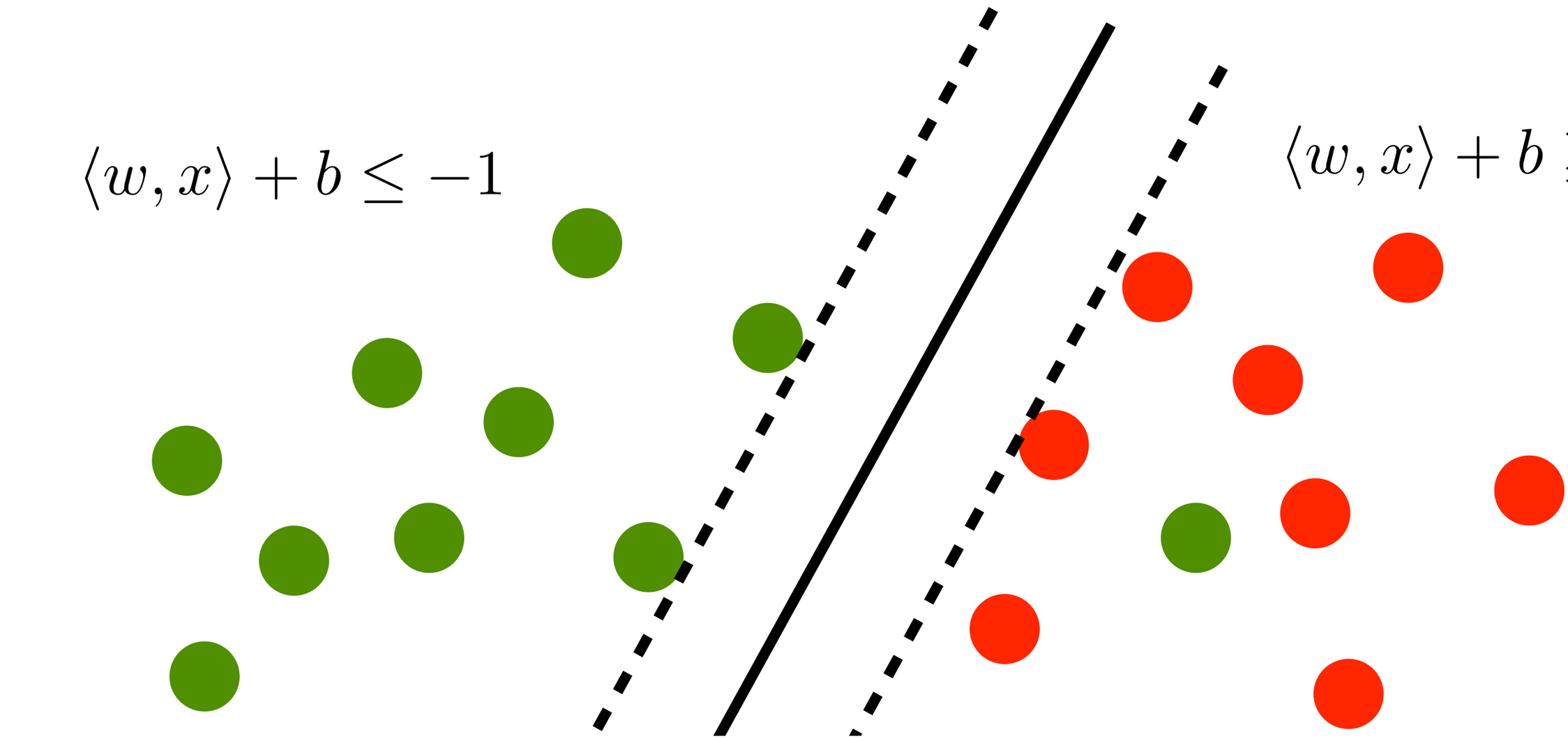
$\langle w, x \rangle + b \leq -1$

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



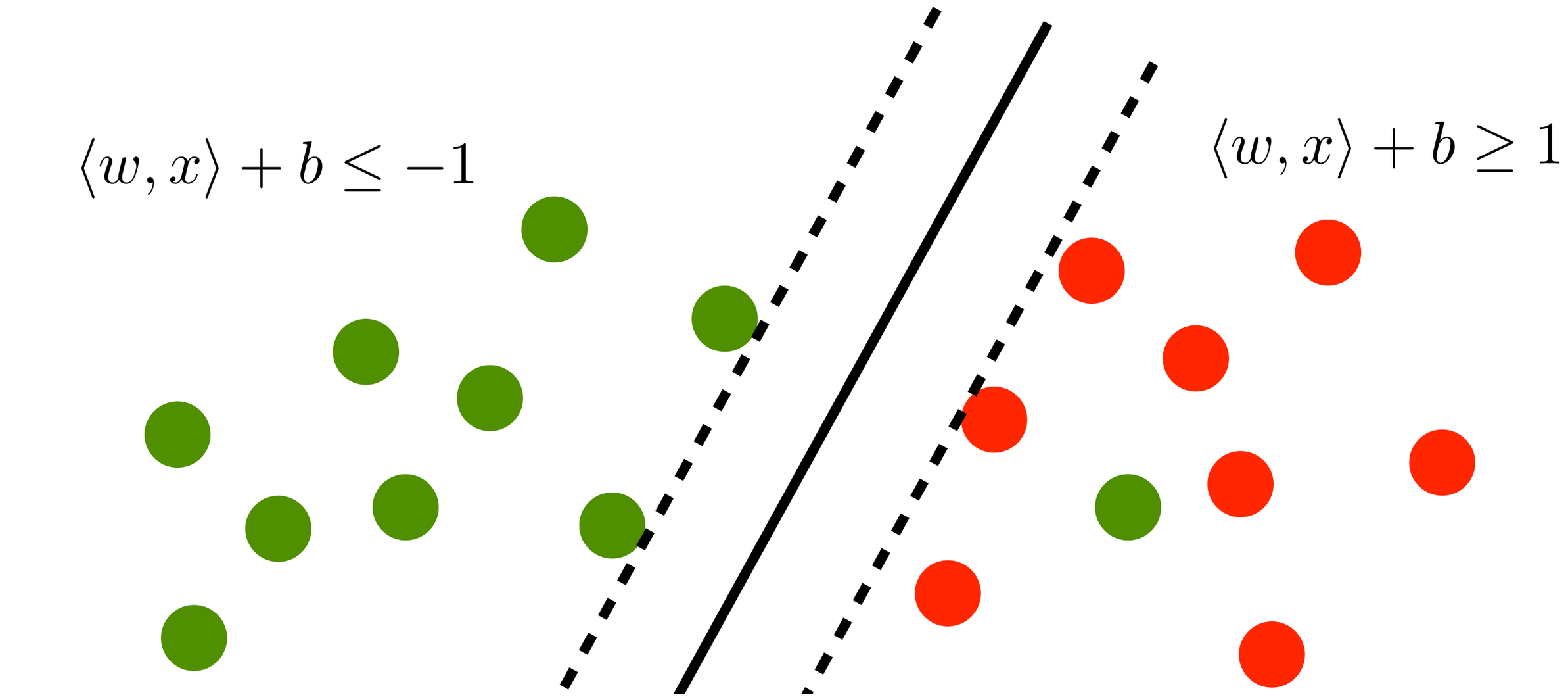
Theorem (Minsky & Papert)
Finding the minimum error separating hyperplane is NP hard

Large Margin Classifier



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

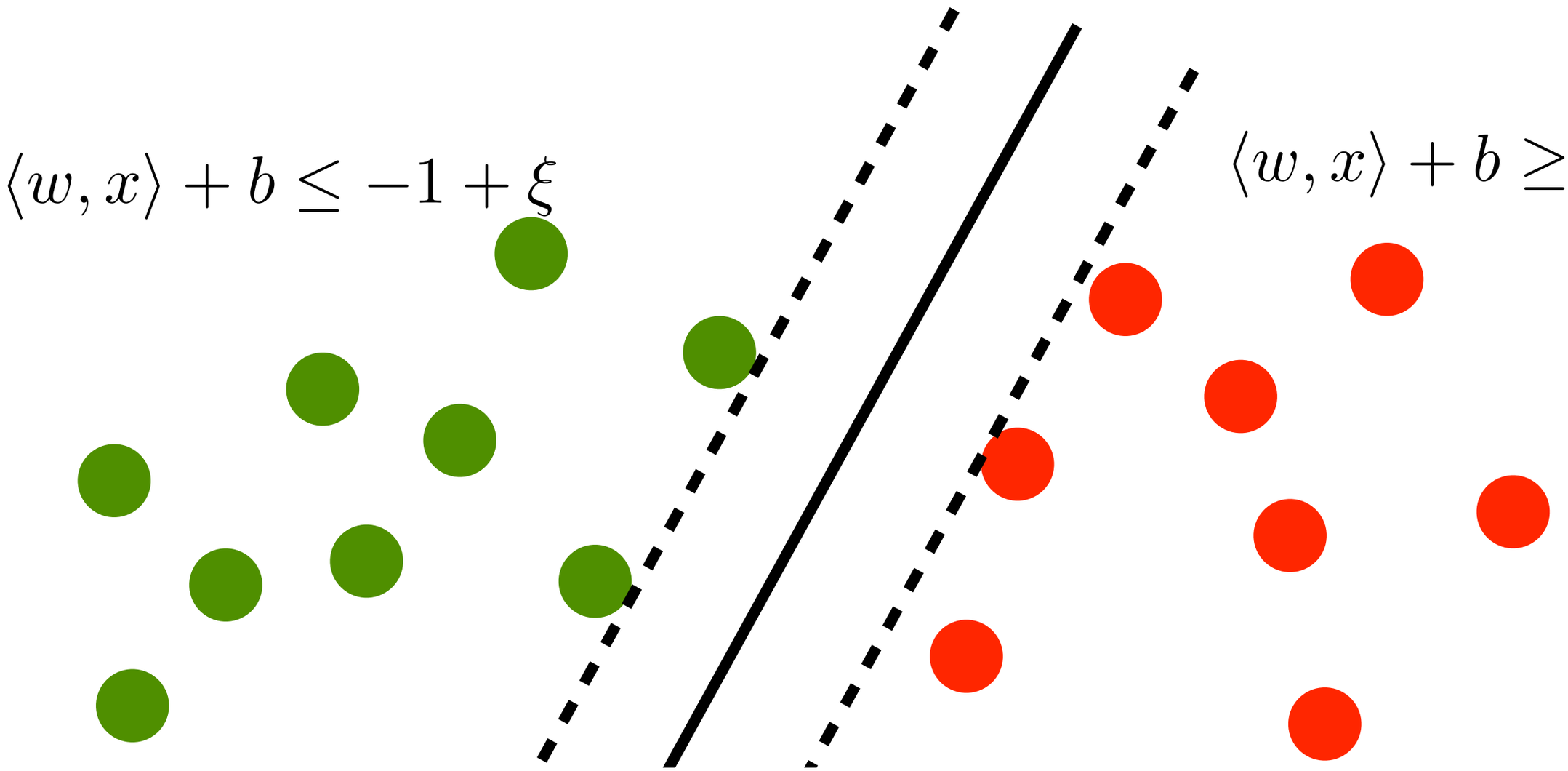
Large Margin Classifier



minimum error separator is impossible

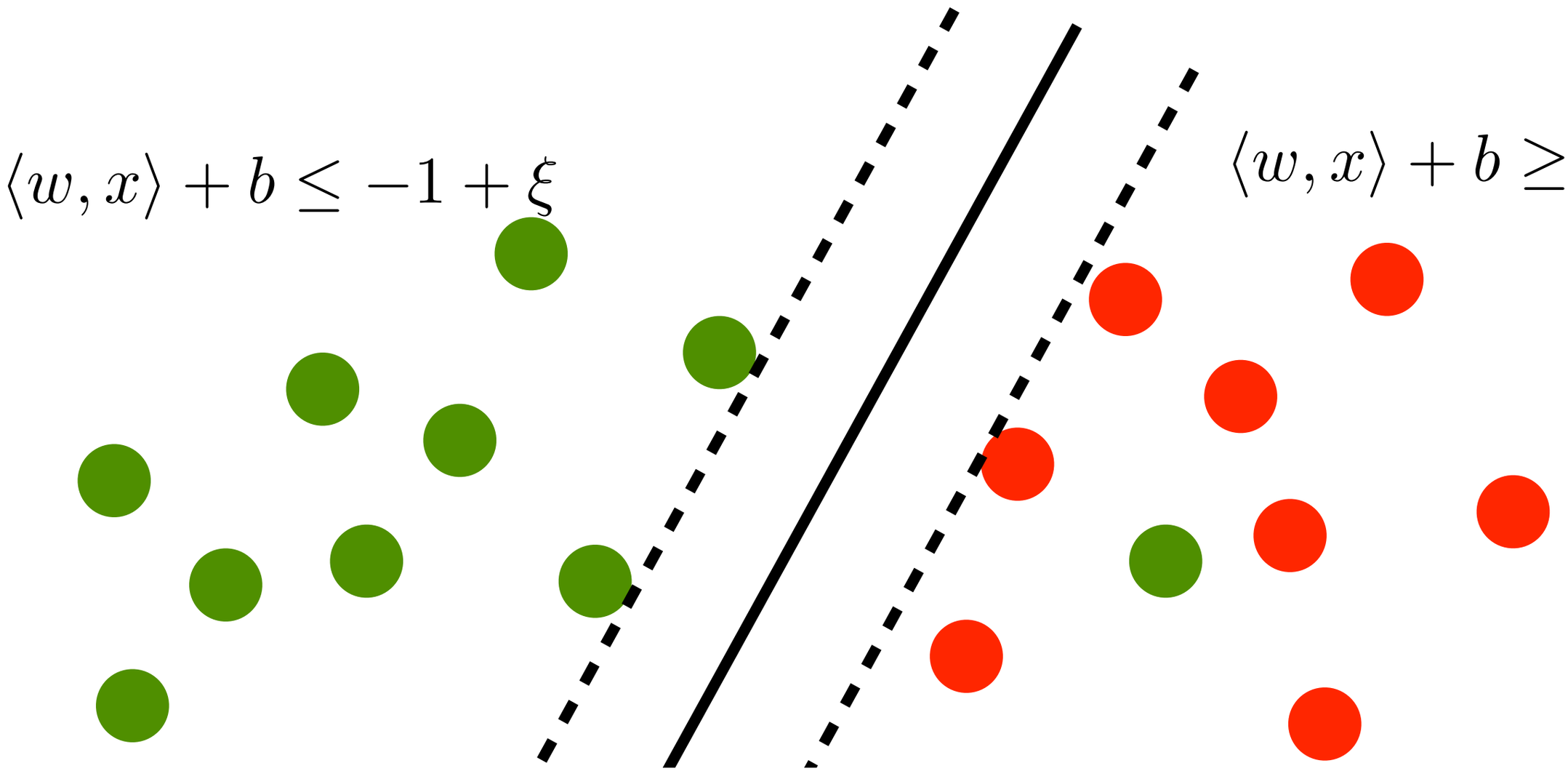
Theorem (Minsky & Papert)
Finding the minimum error separating hyperplane is NP hard

Adding slack variables



Convex optimization problem

Adding slack variables

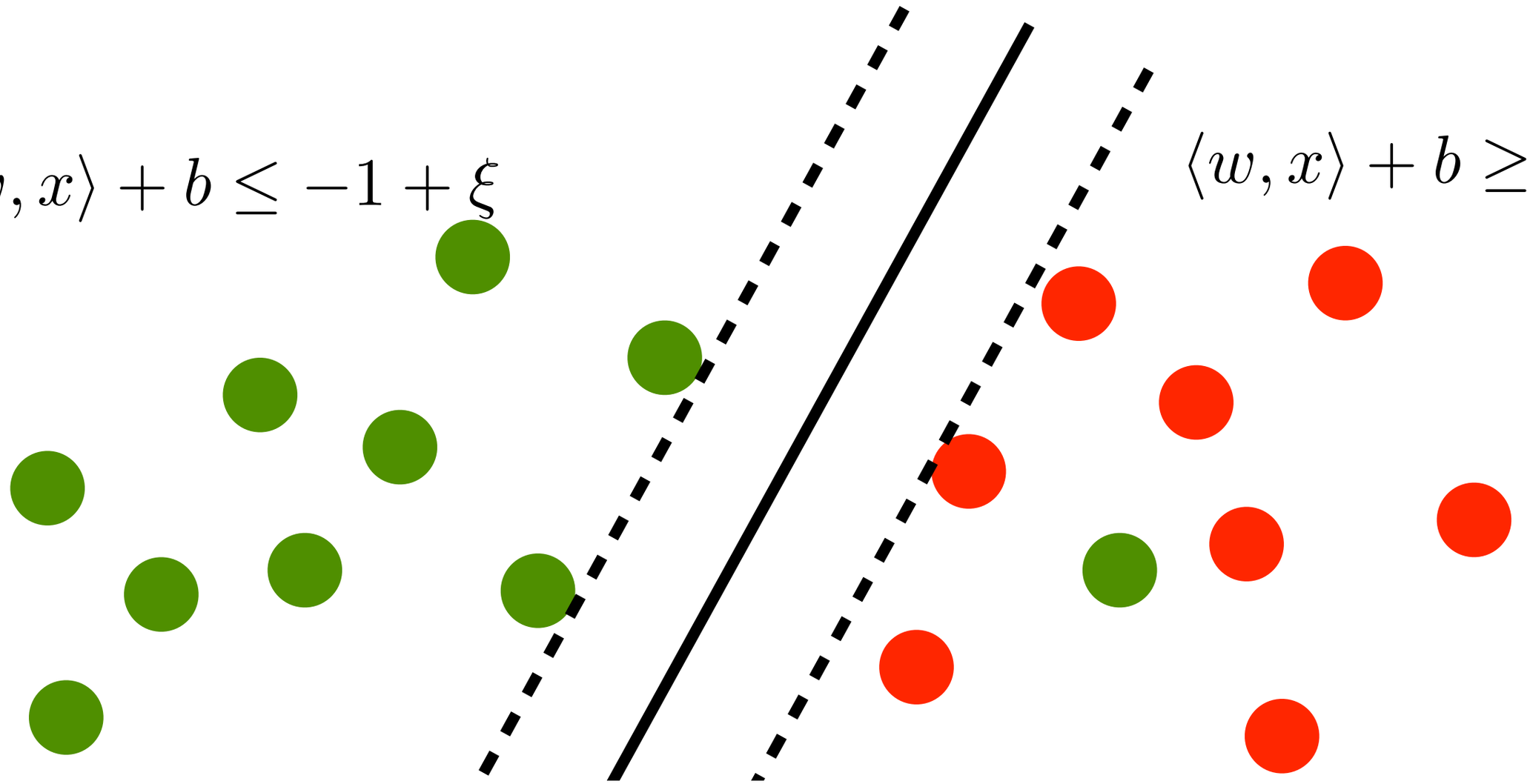


Convex optimization problem

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. $w = 0$ and $b = 0$ and $\xi_i = 1$

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,b,ξ is at saddle point with α,η

- Derivatives in w,b,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, b 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

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

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

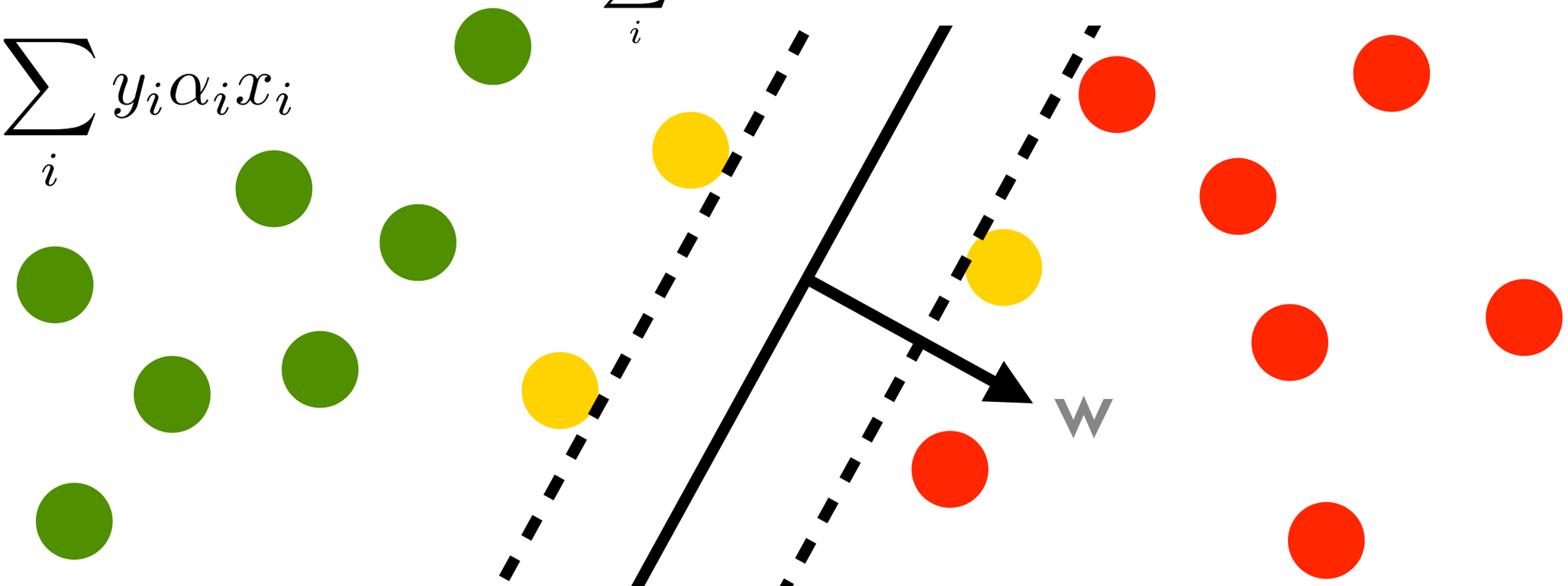
bound
influence

Karush Kuhn Tucker Conditions

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

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

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



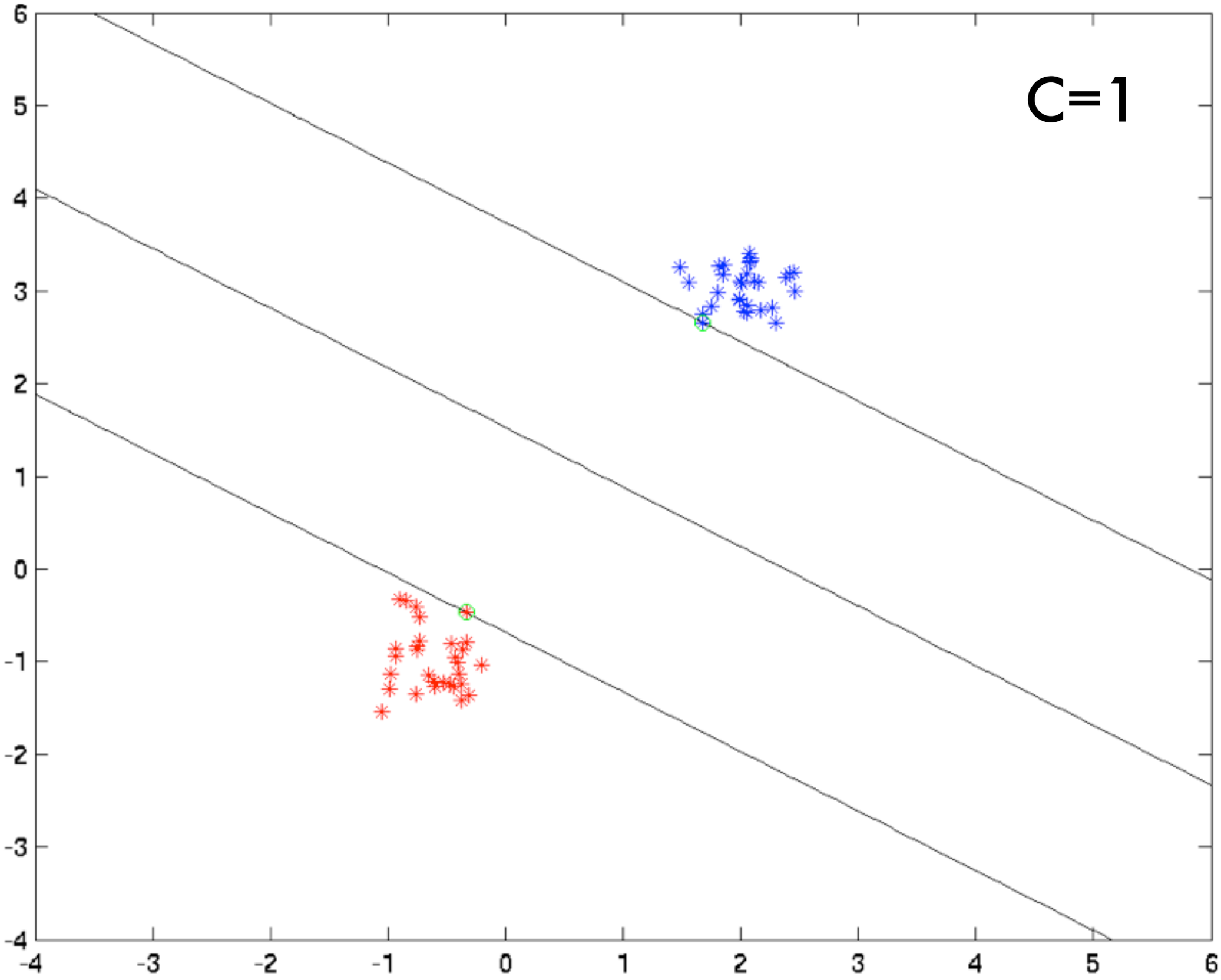
$$\alpha_i \left[y_i \left[\langle w, x_i\rangle + b\right] + \xi_i - 1\right] = 0$$

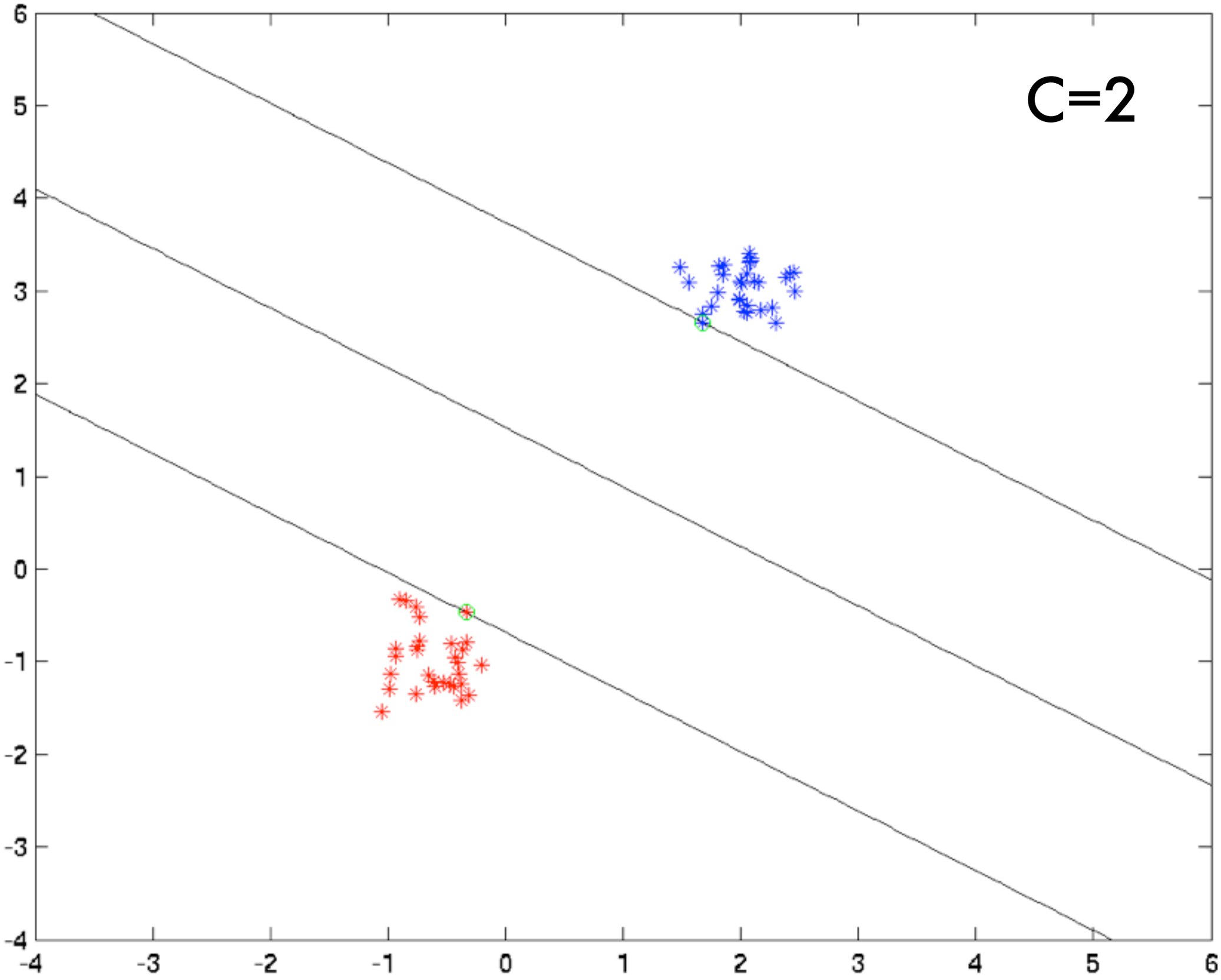
$$\eta_i \xi_i = 0$$

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

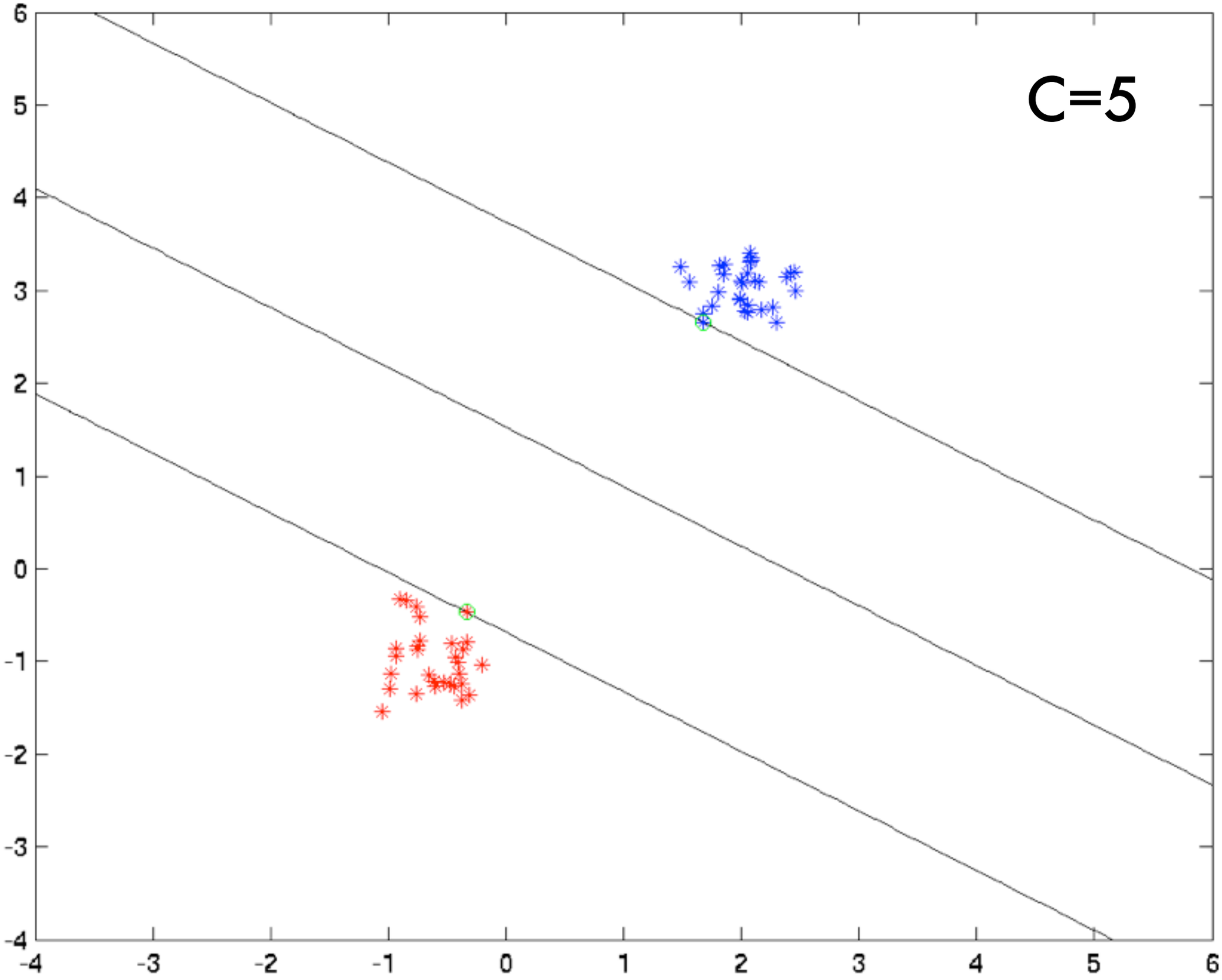
$$0 < \alpha_i < C \implies y_i \left[\langle w, x_i\rangle + b\right] = 1$$

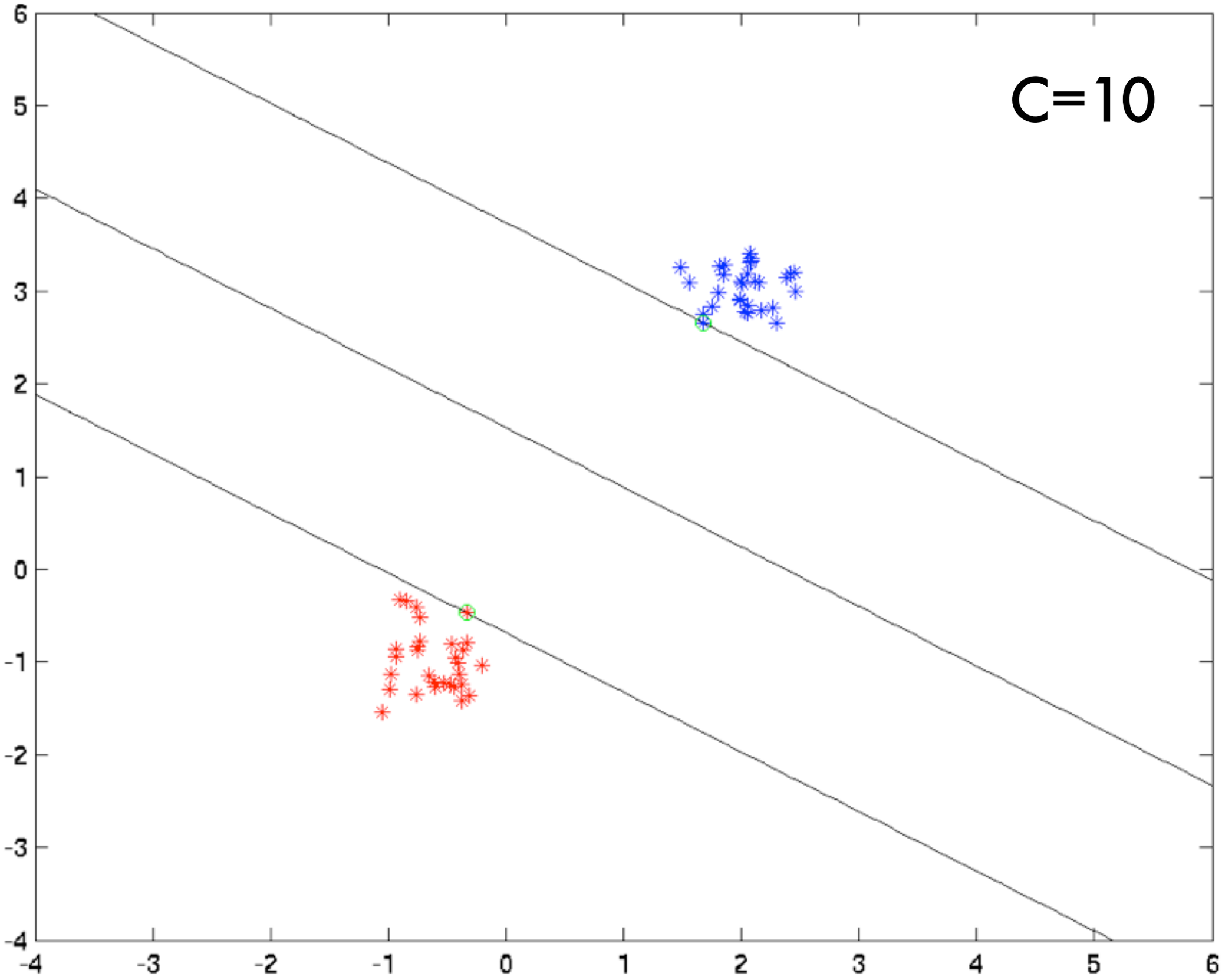
$$\alpha_i = C \implies y_i \left[\langle w, x_i\rangle + b\right] \leq 1$$

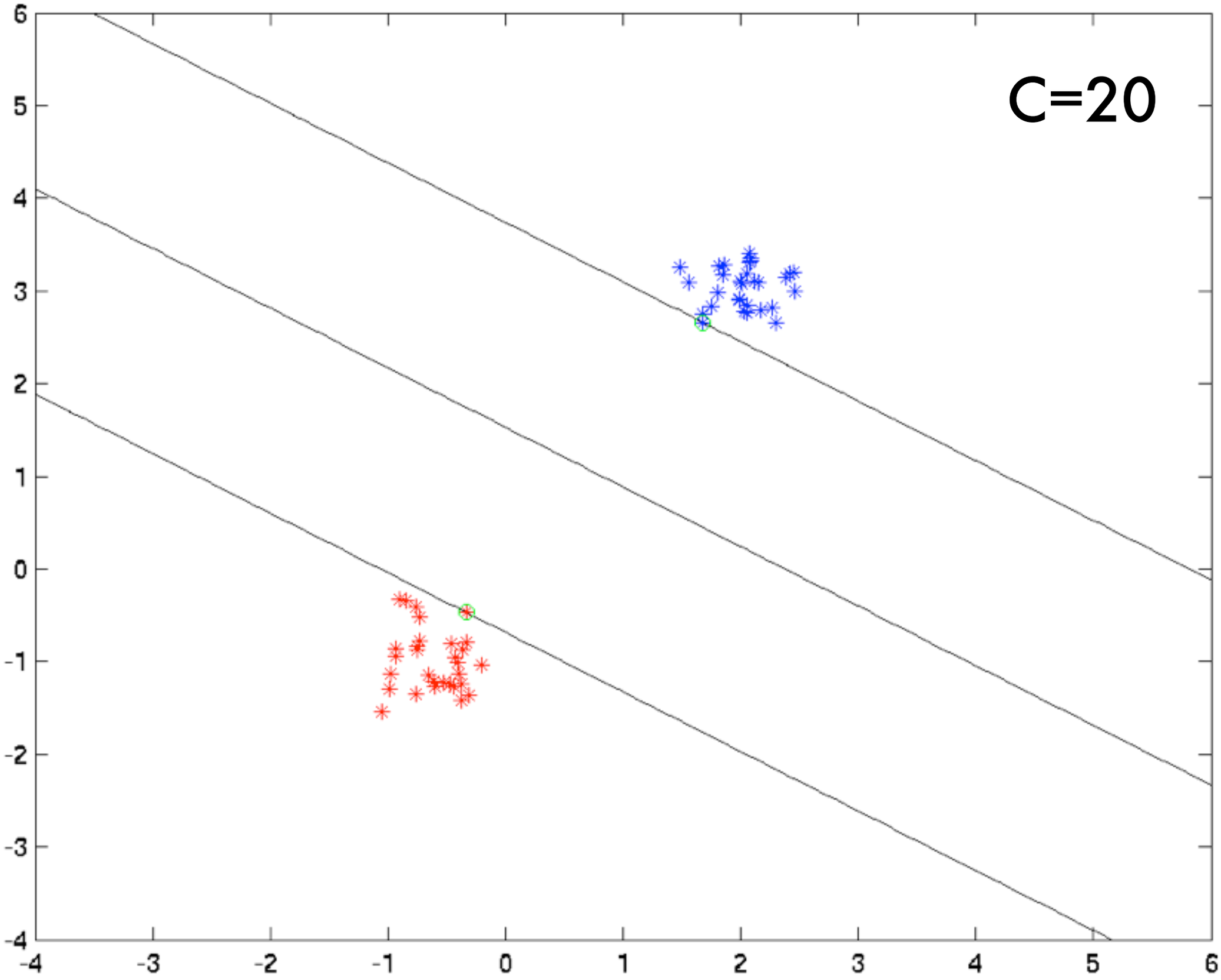

C=1
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

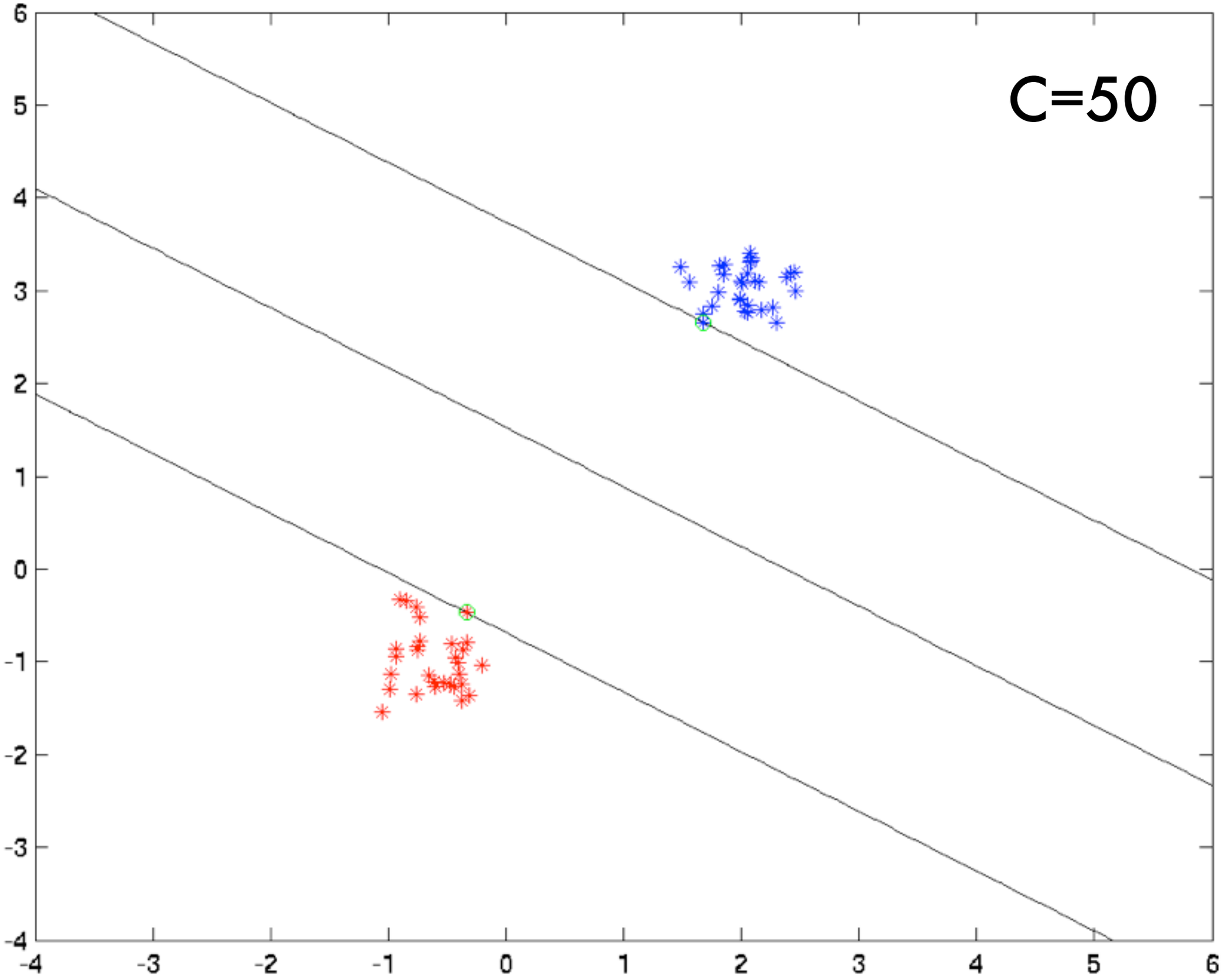

C=2
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

C=5

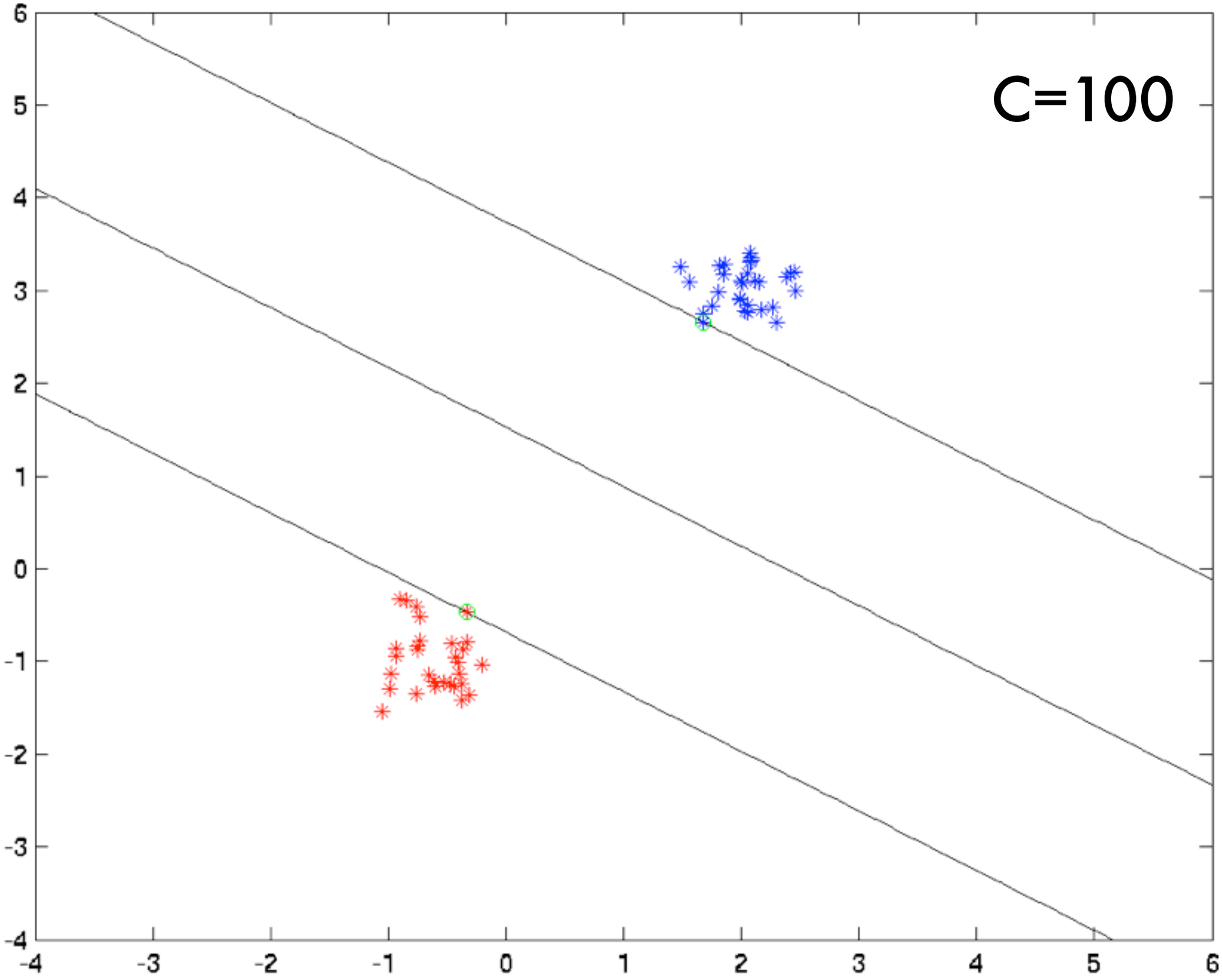


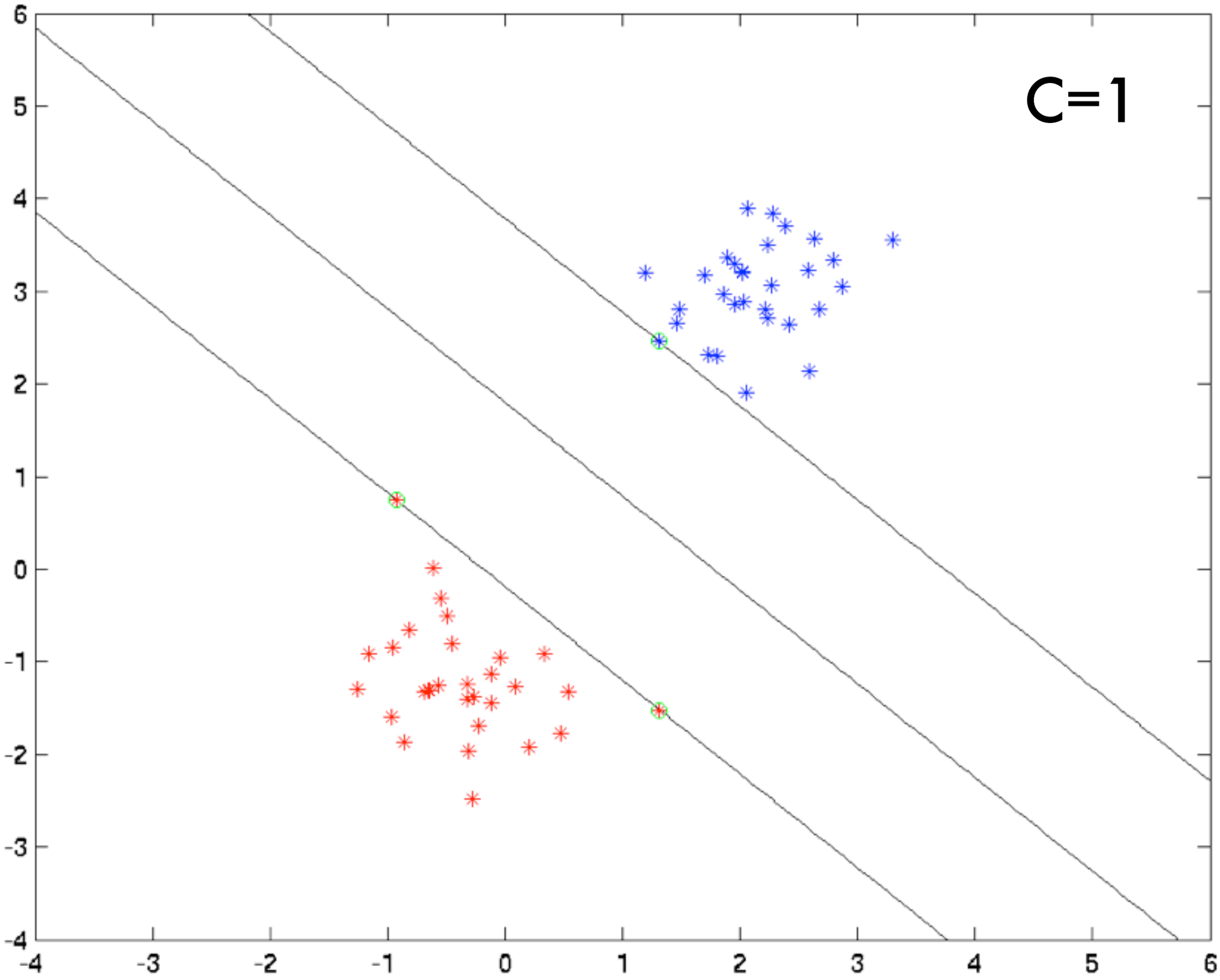

C=10
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

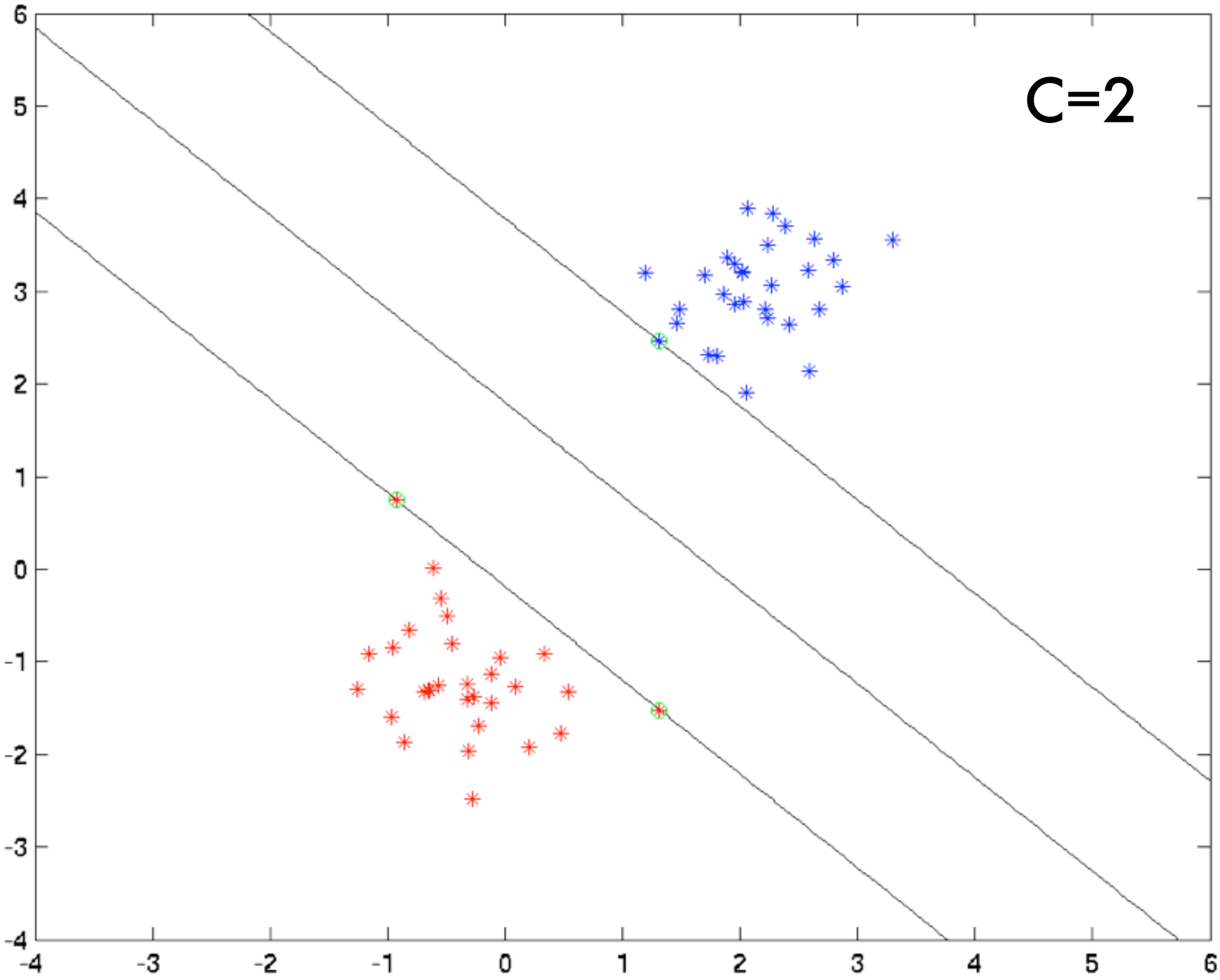

C=20
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

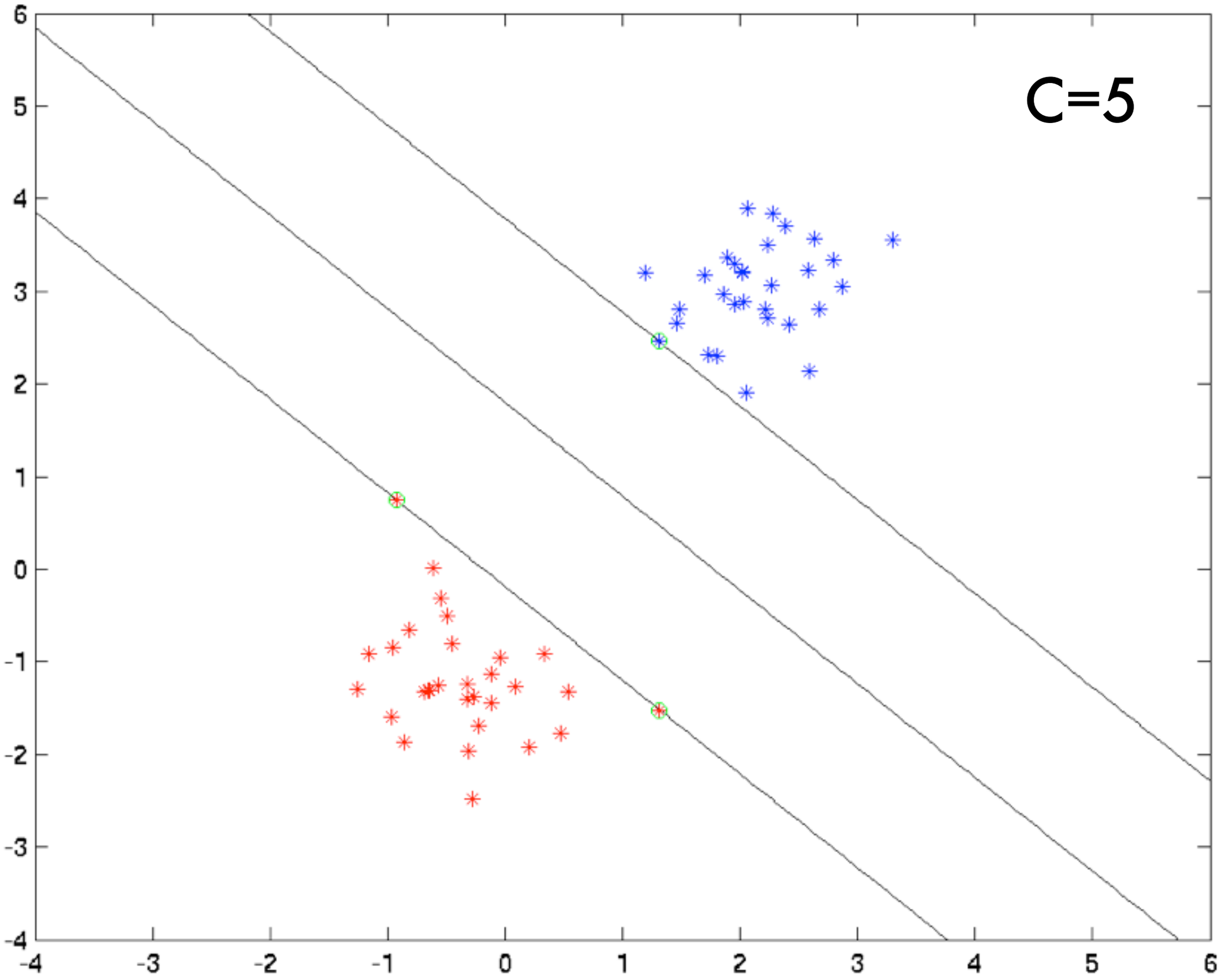

C=50
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

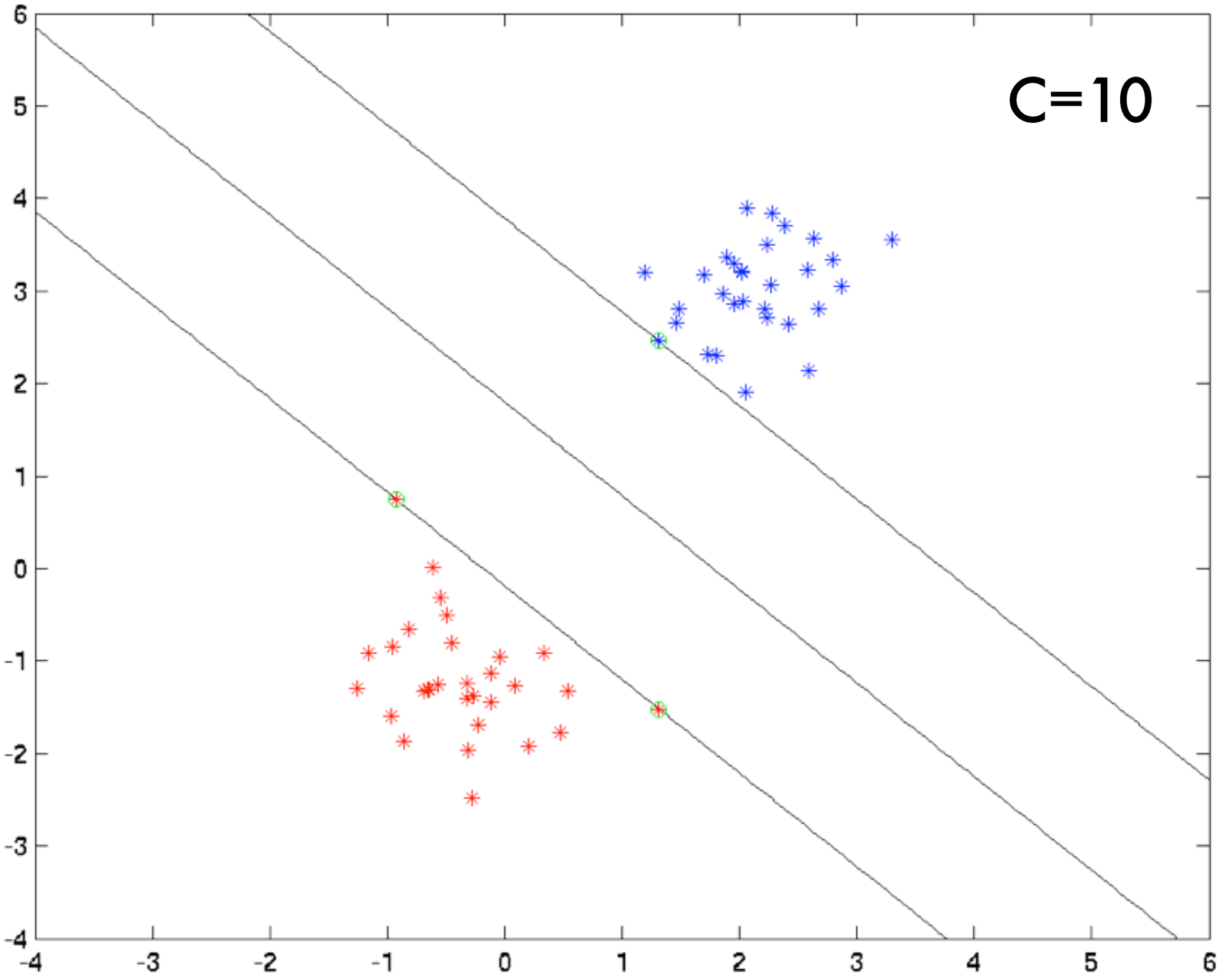
C=100

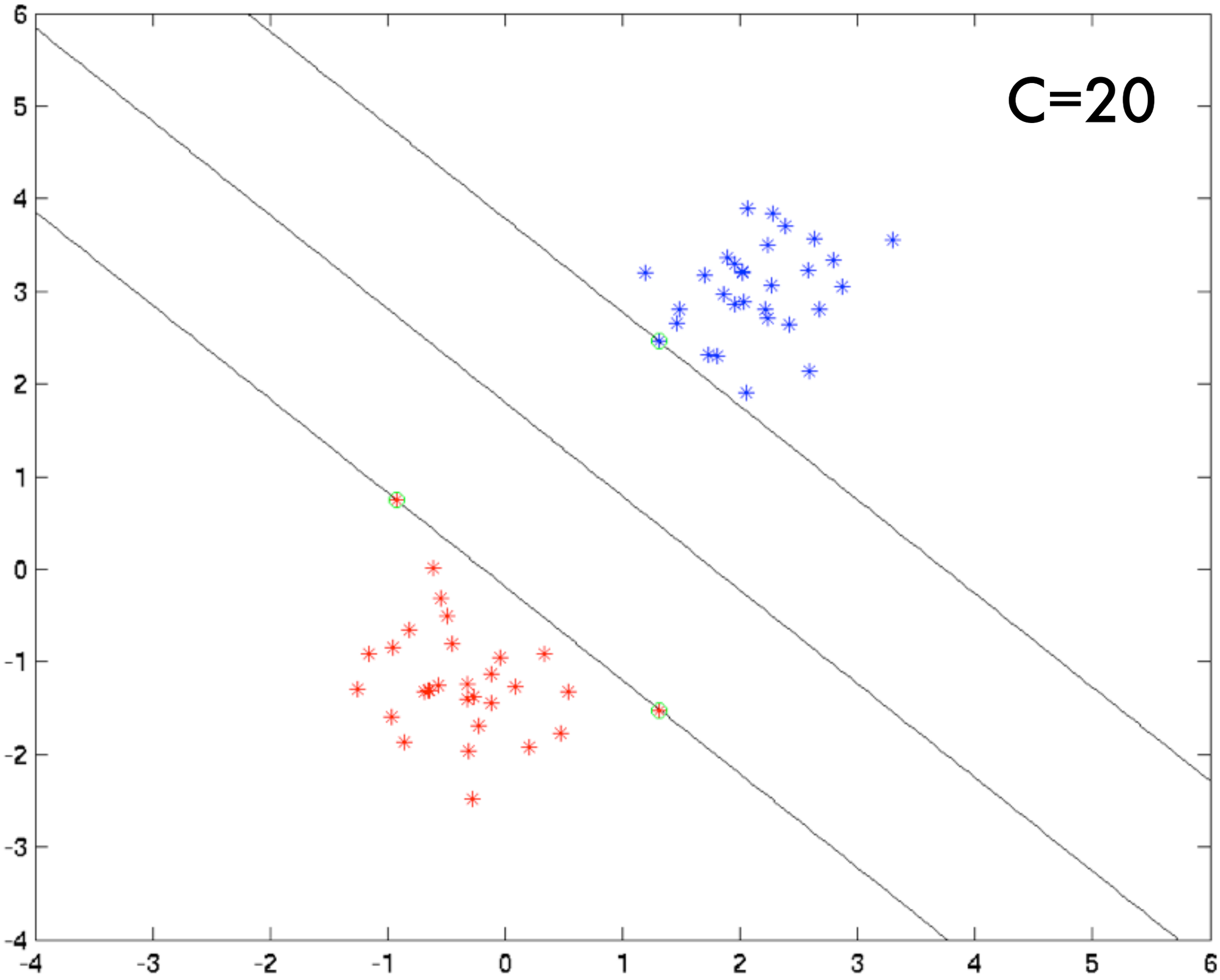


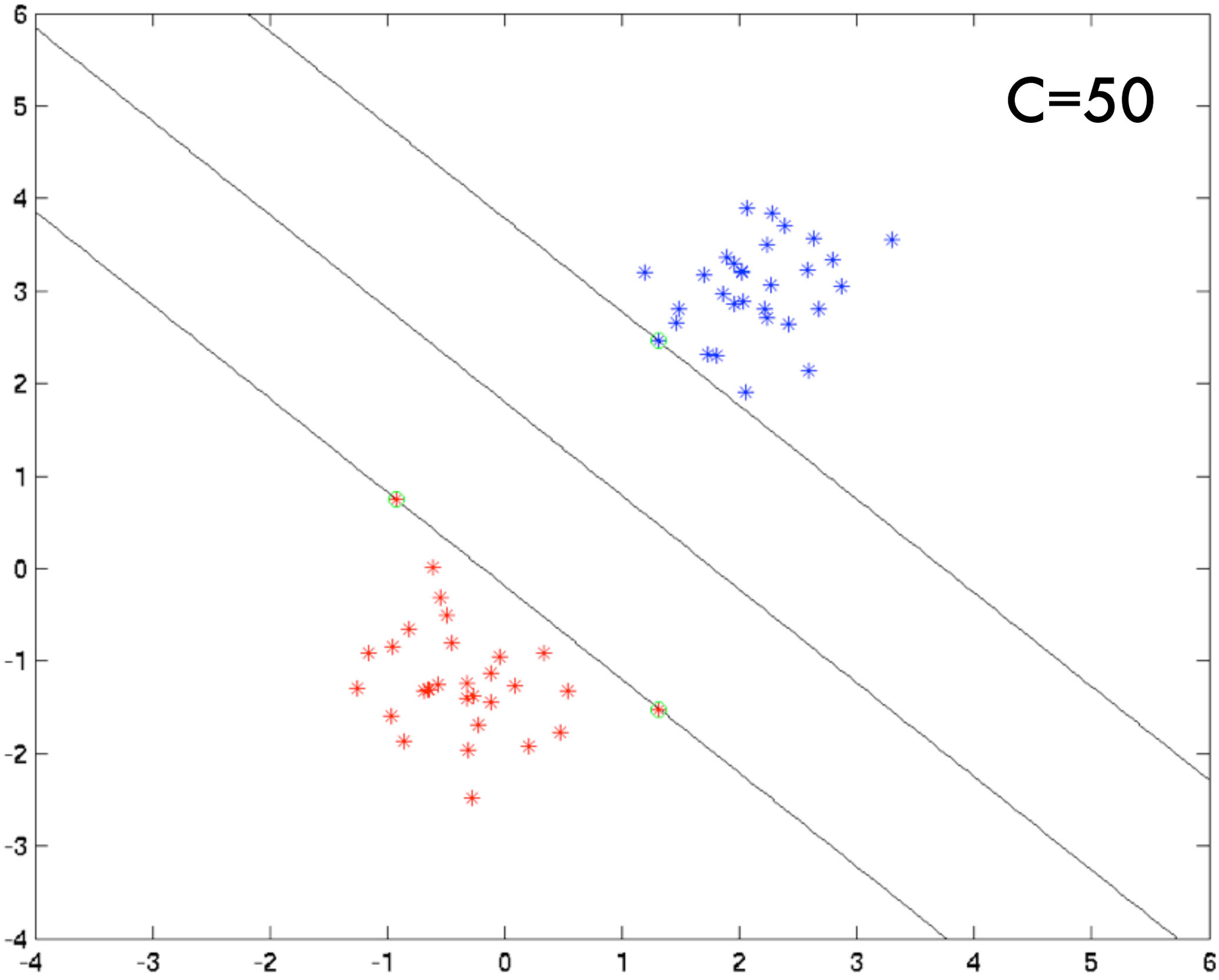

C=1
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

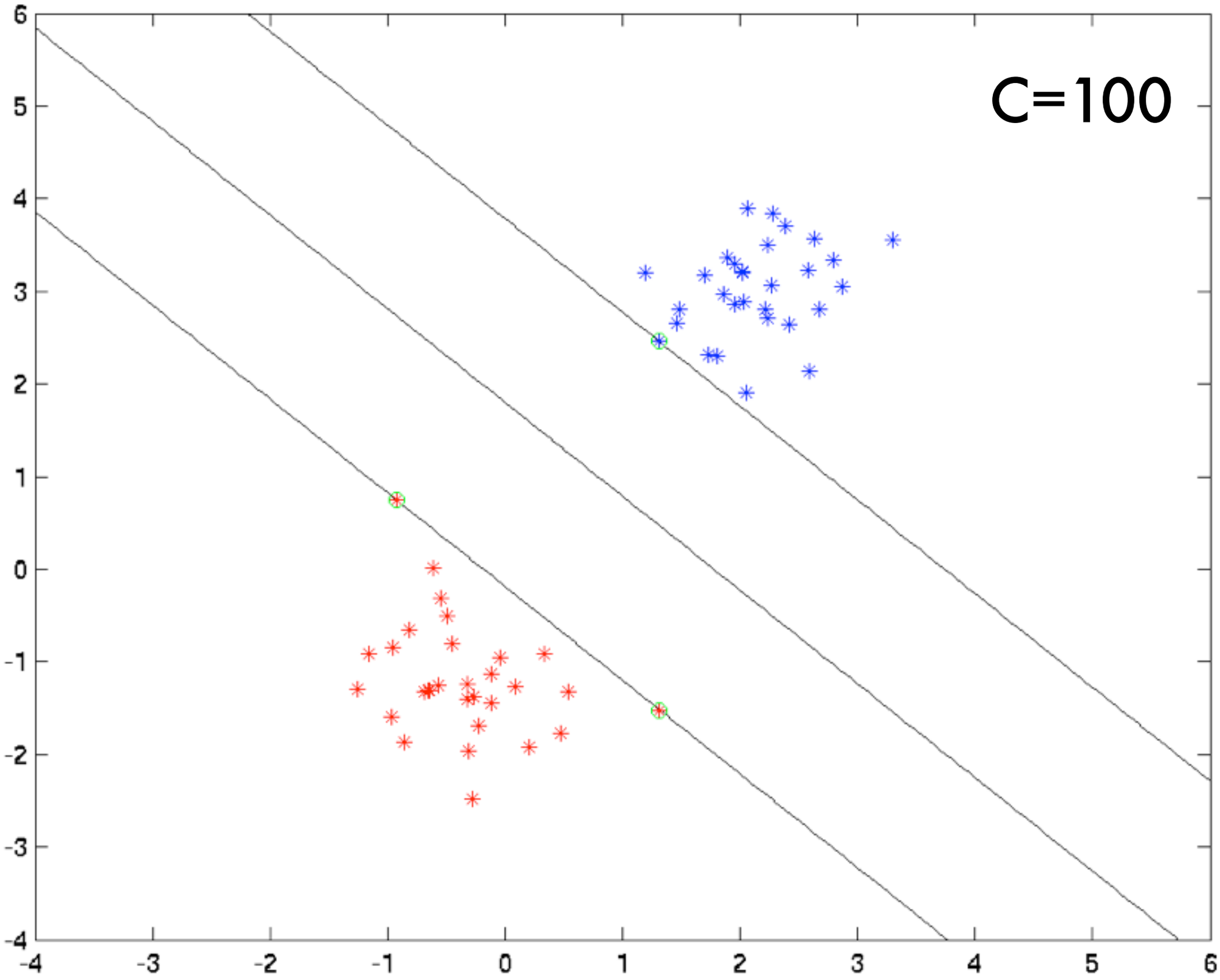

C=2
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

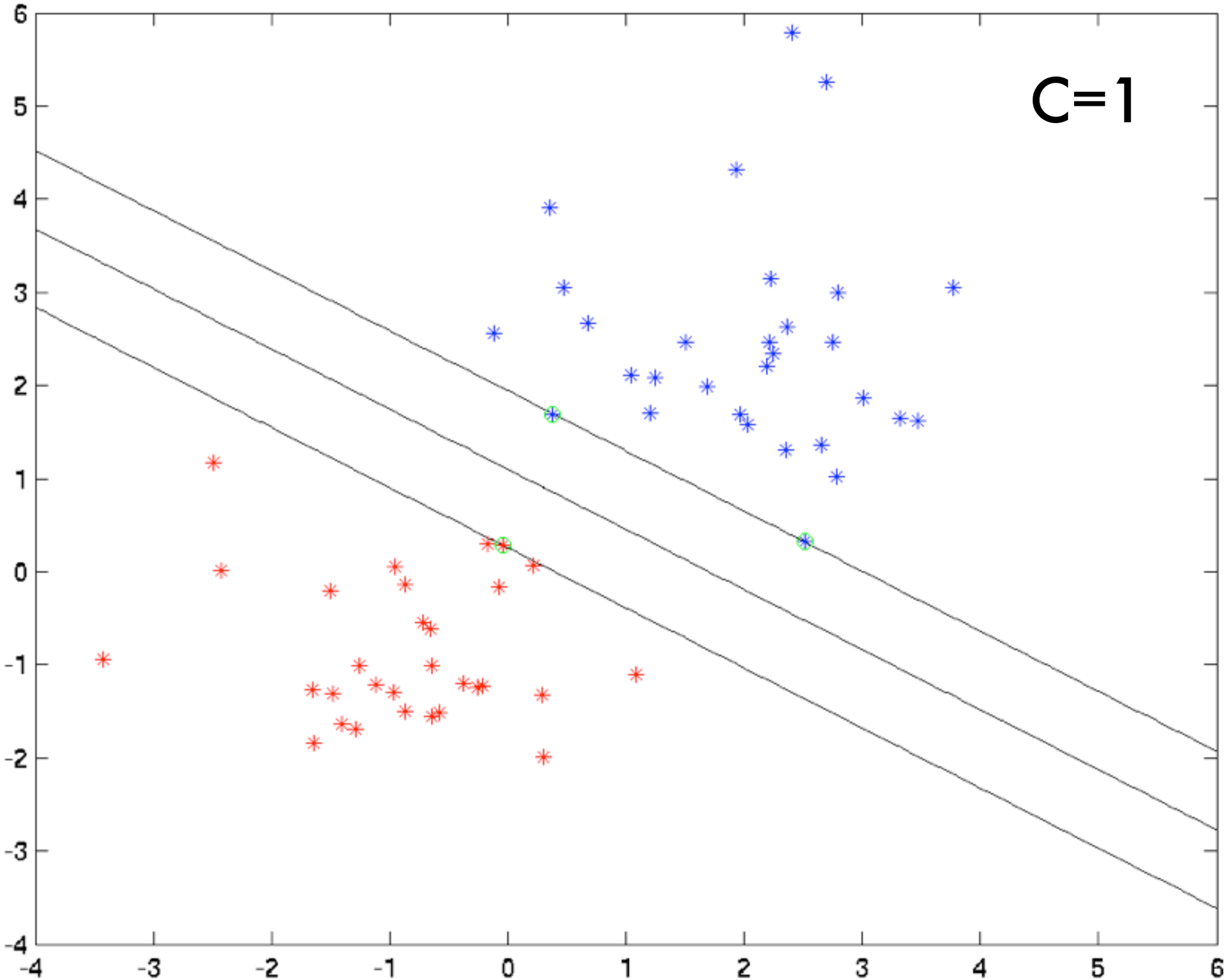

C=5
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

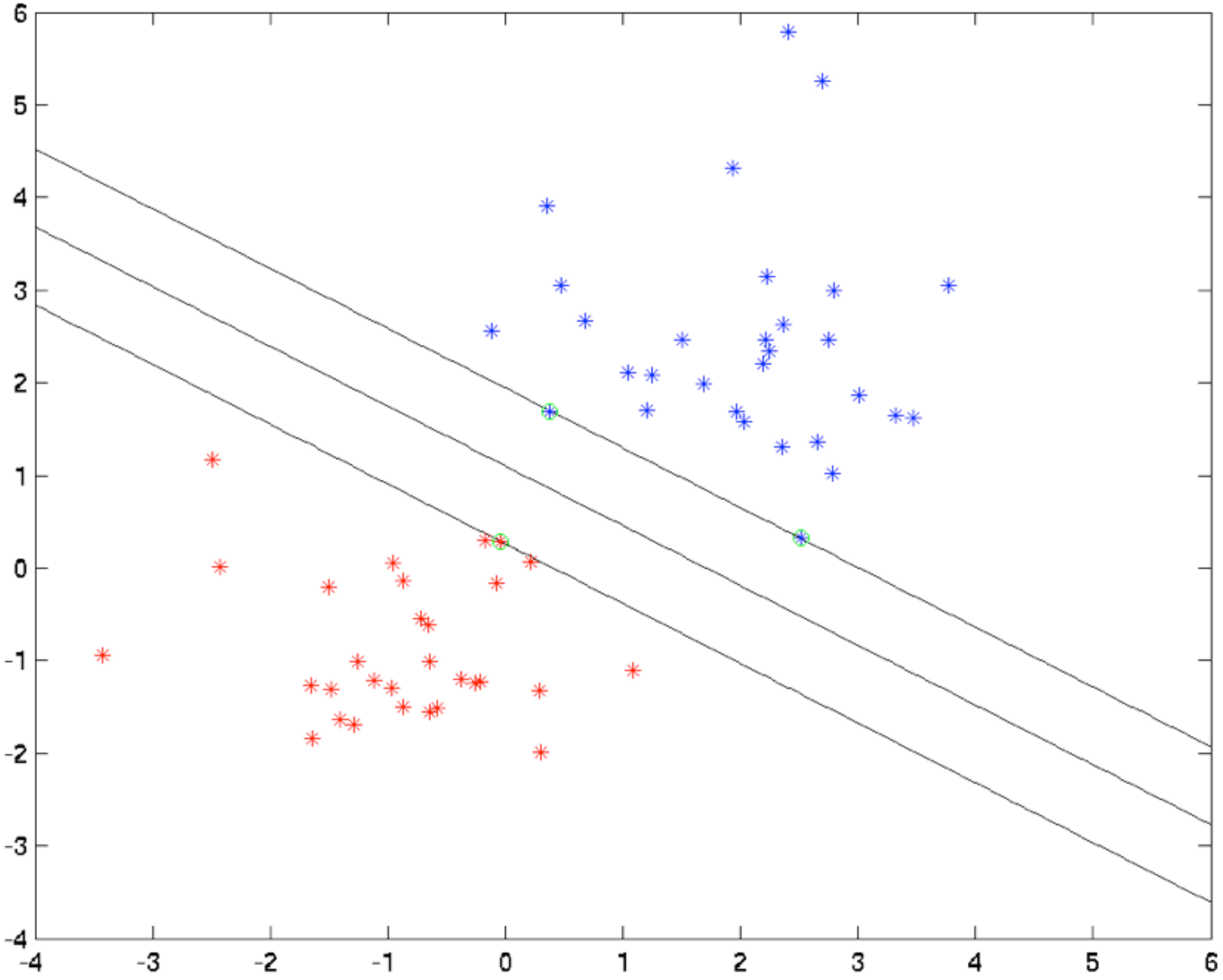

C=10
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

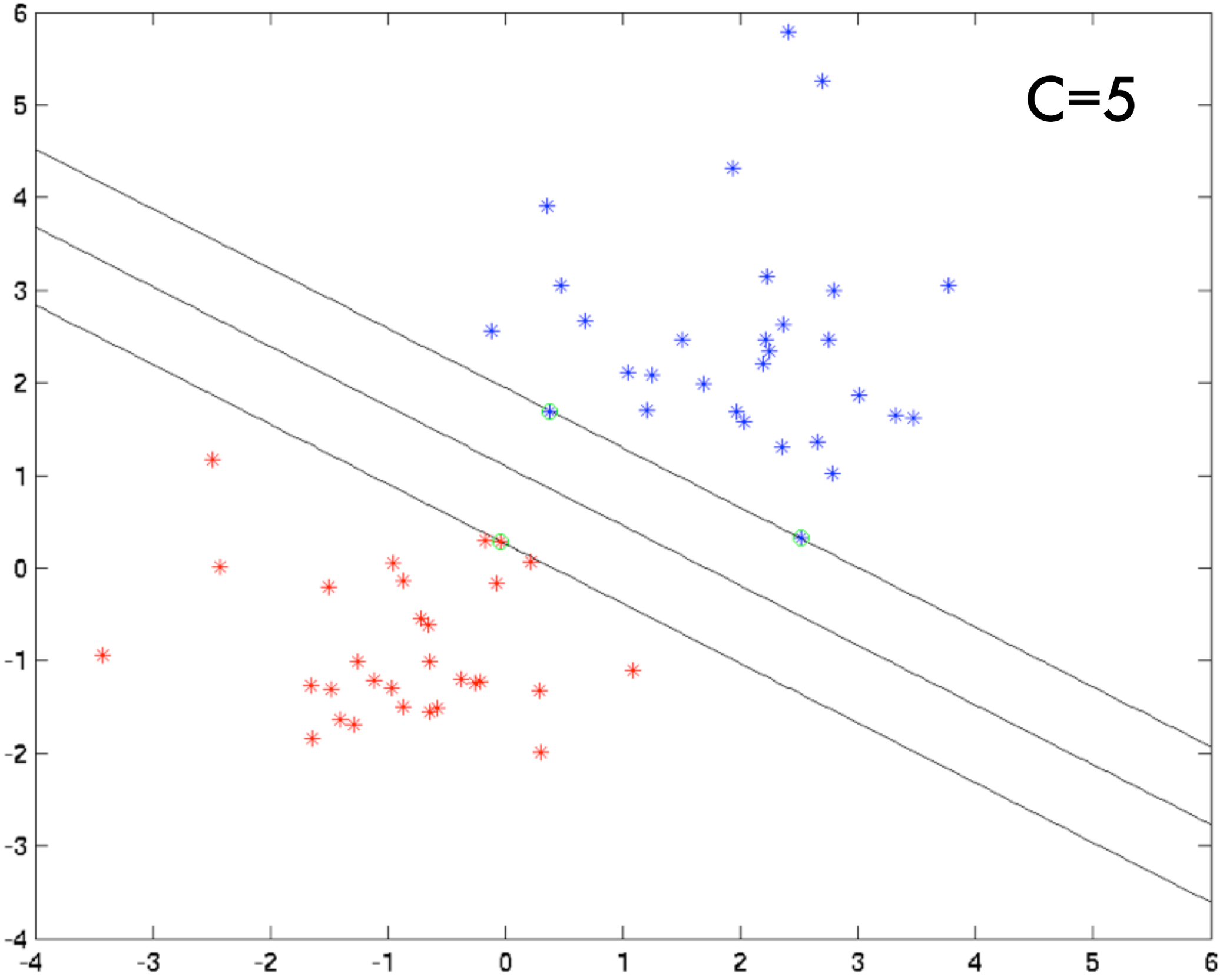

C=20
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

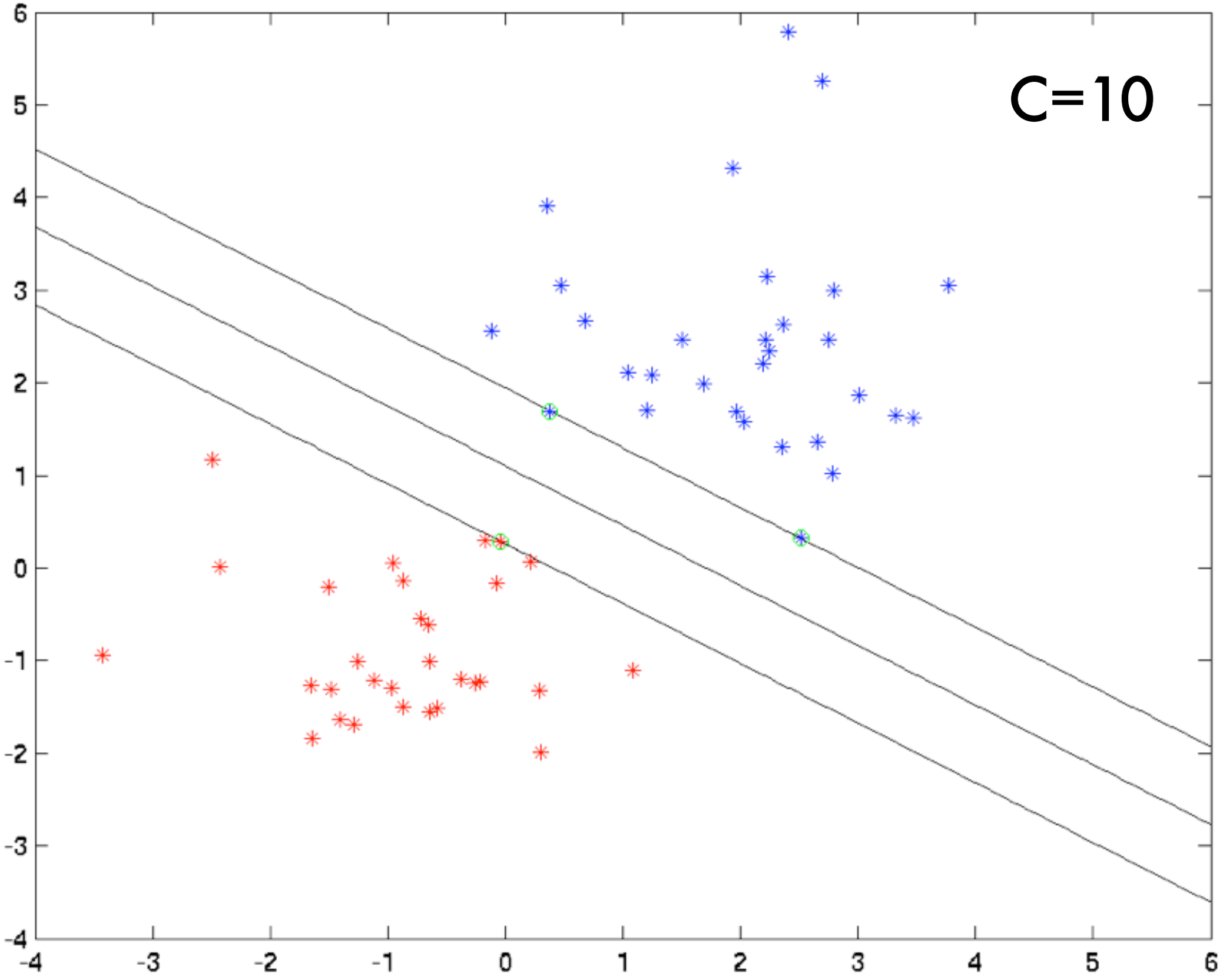

C=50
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

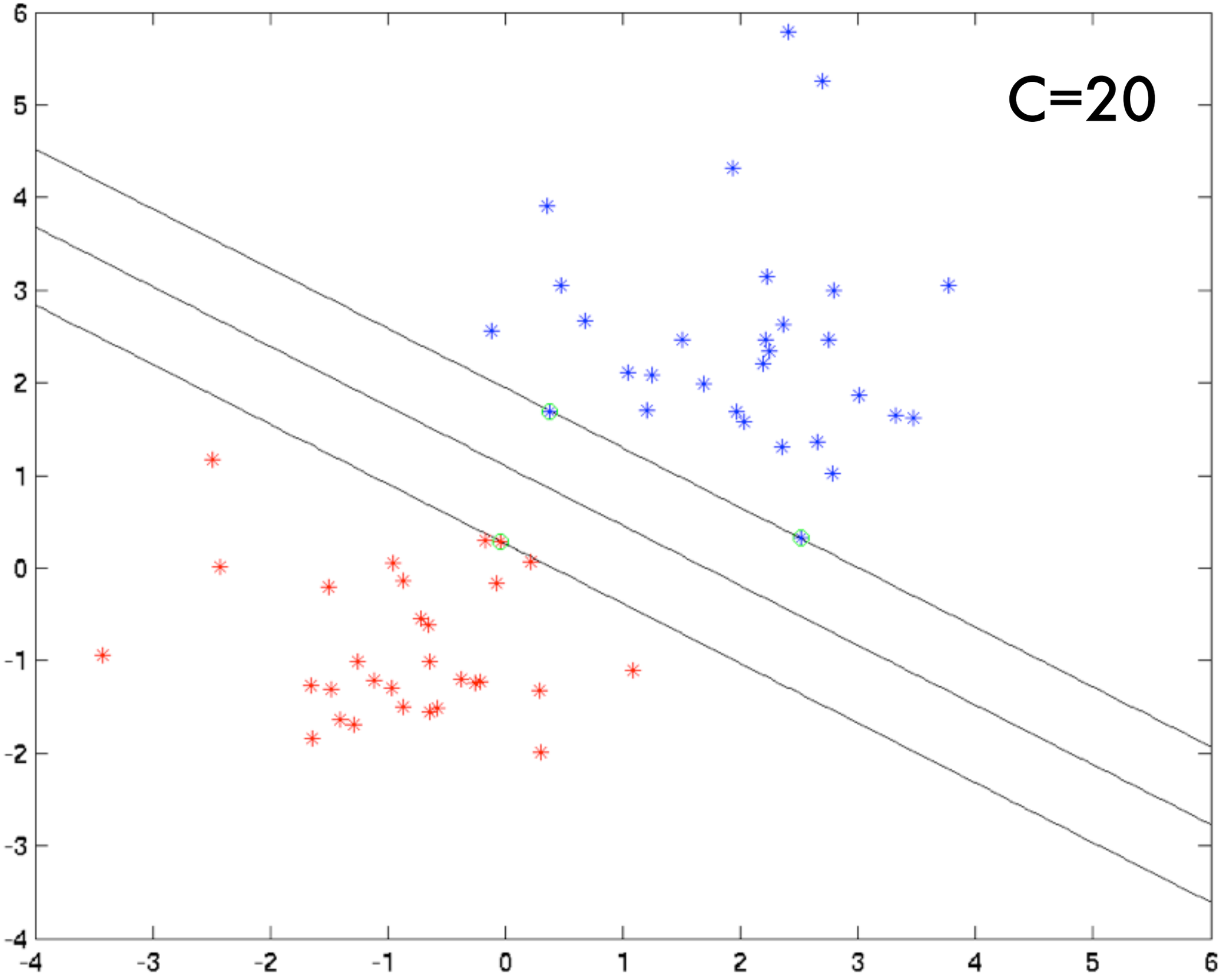

C=100
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

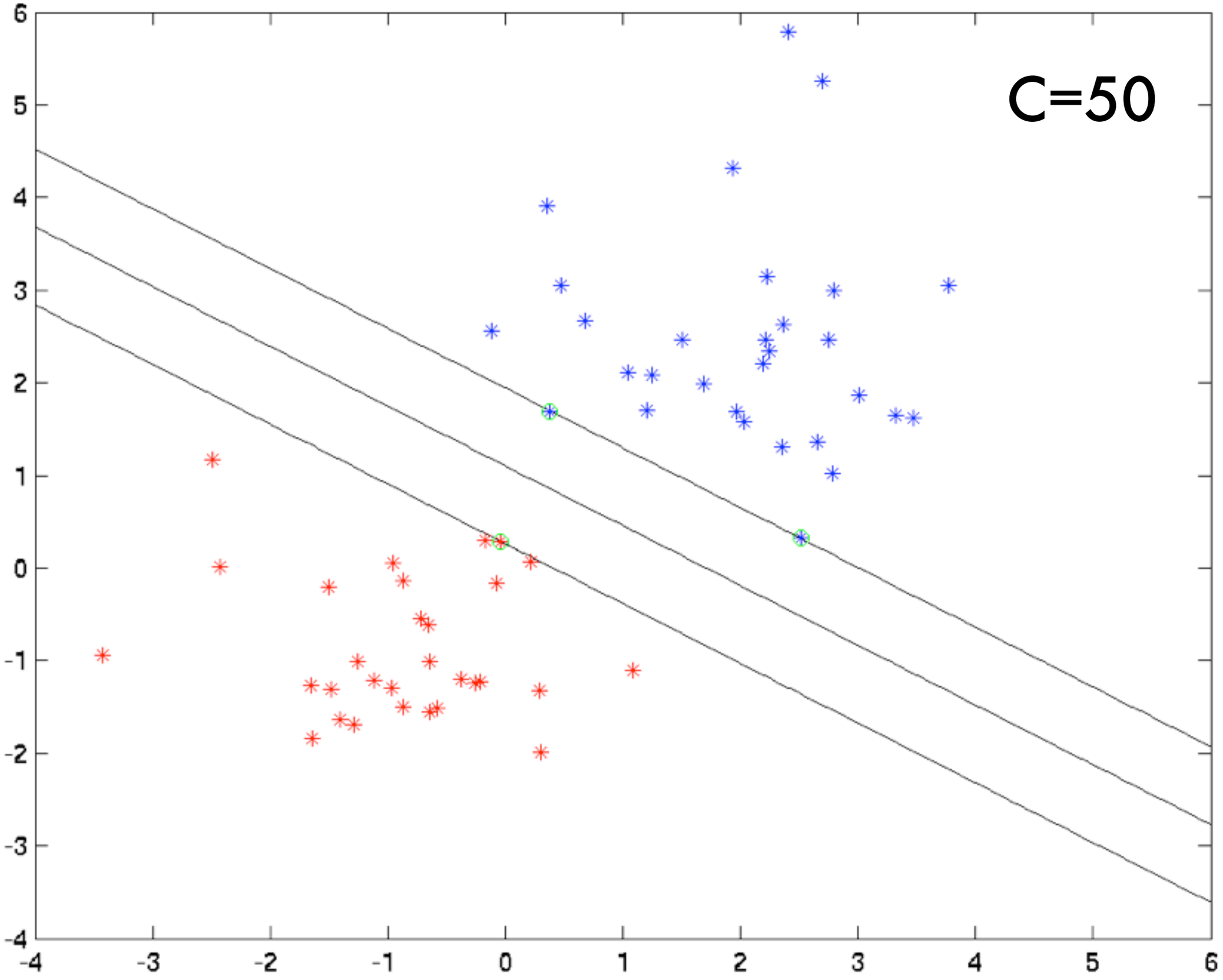

C=1
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

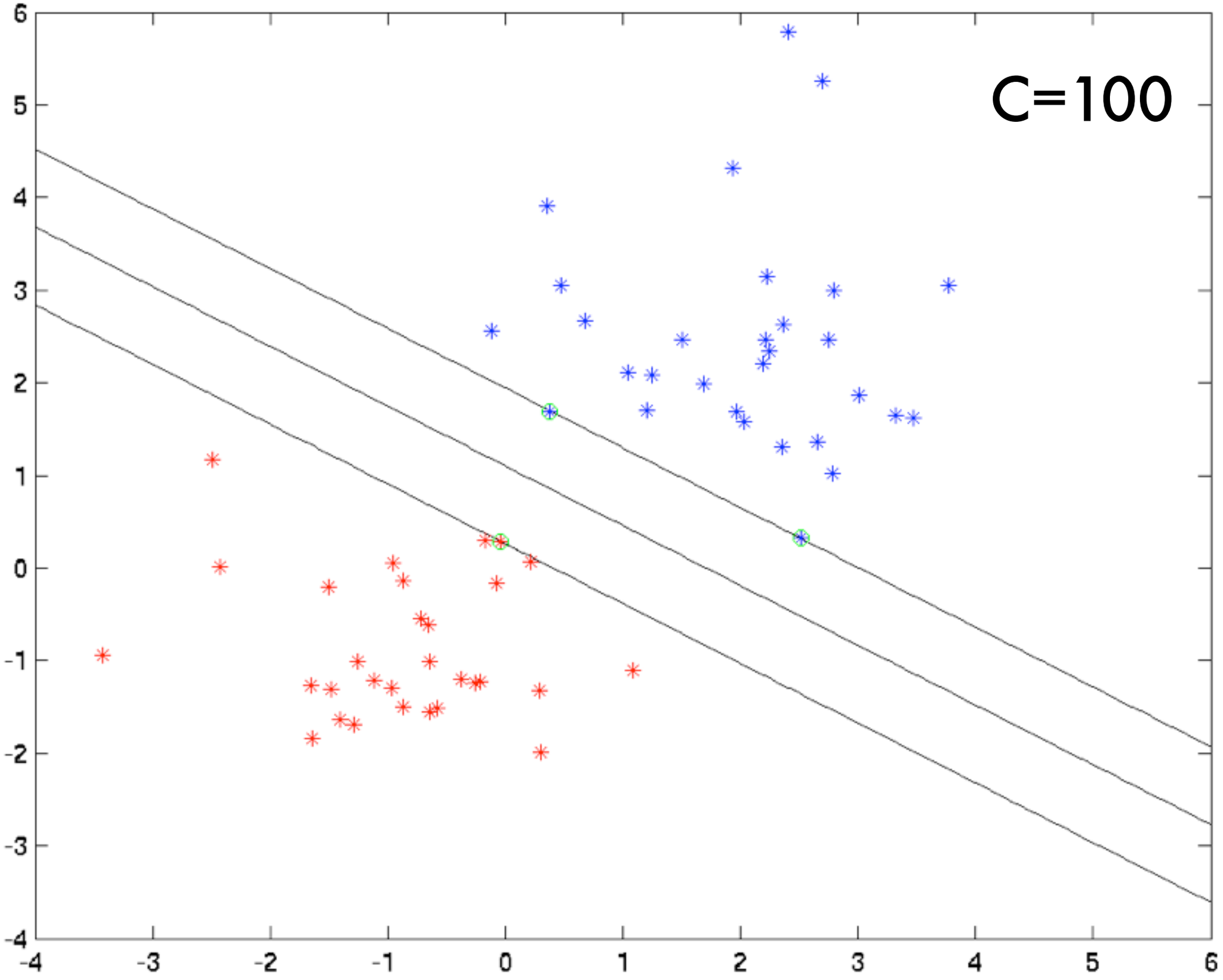

6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

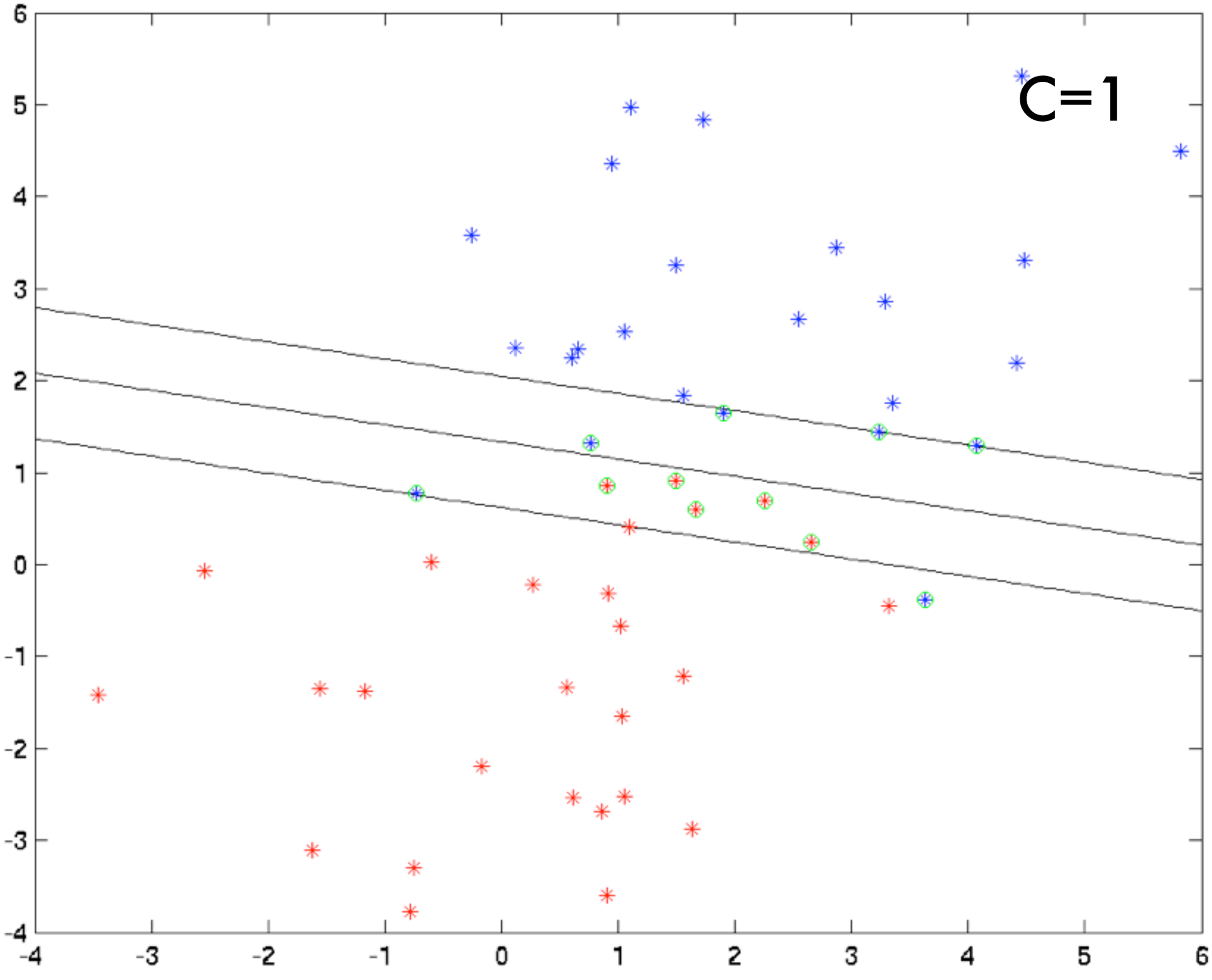

C=5
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

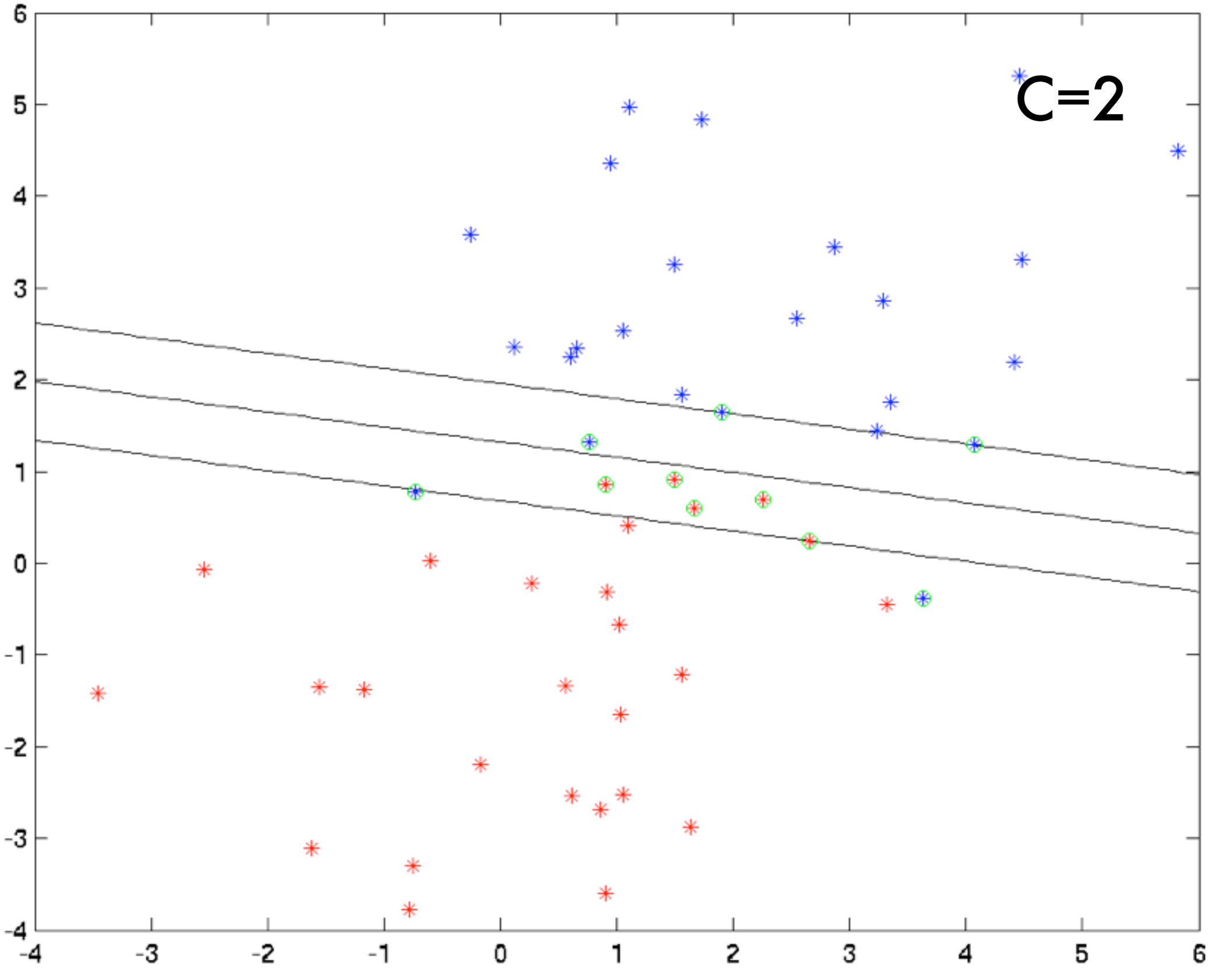

C=10
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

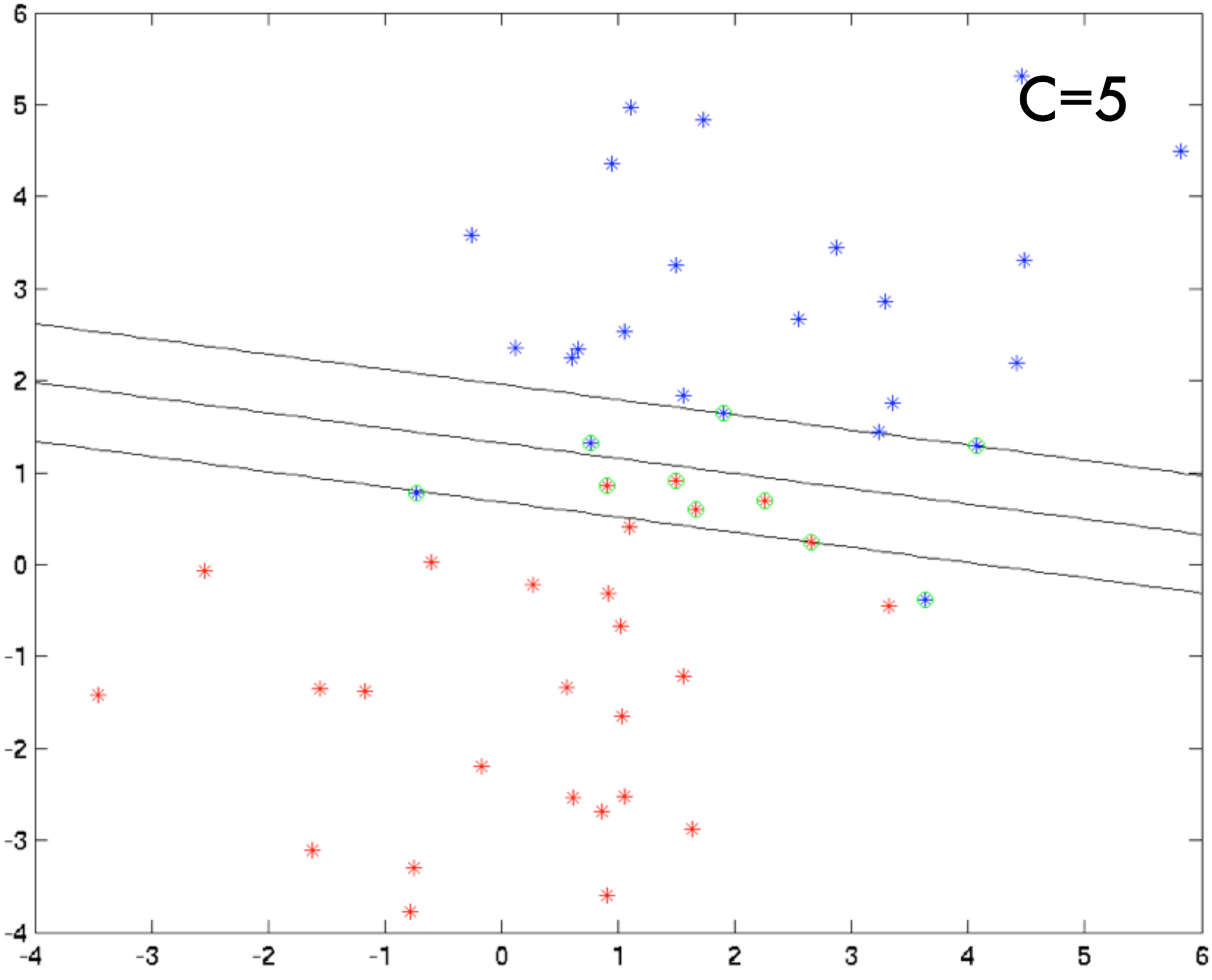

C=20
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

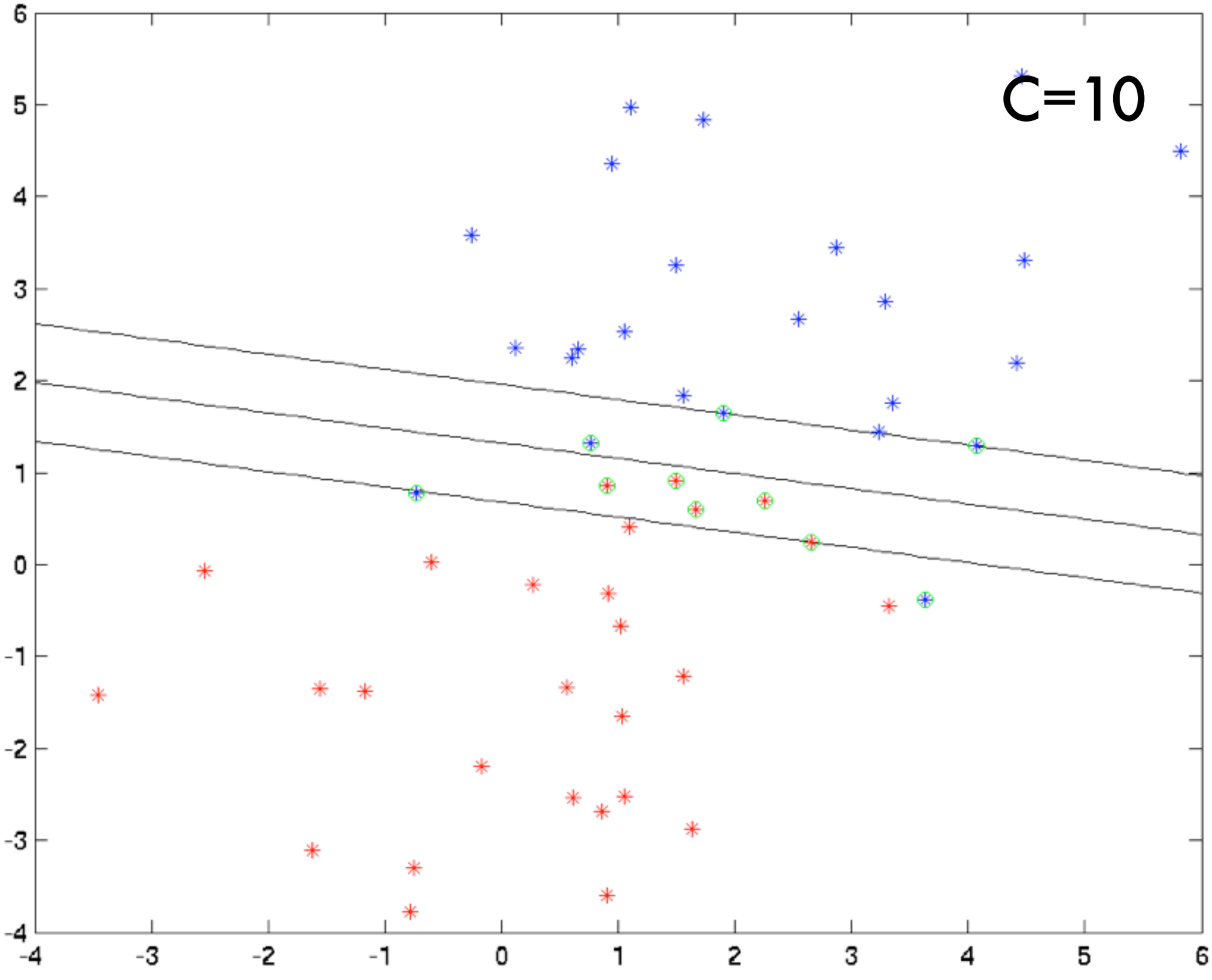

C=50
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

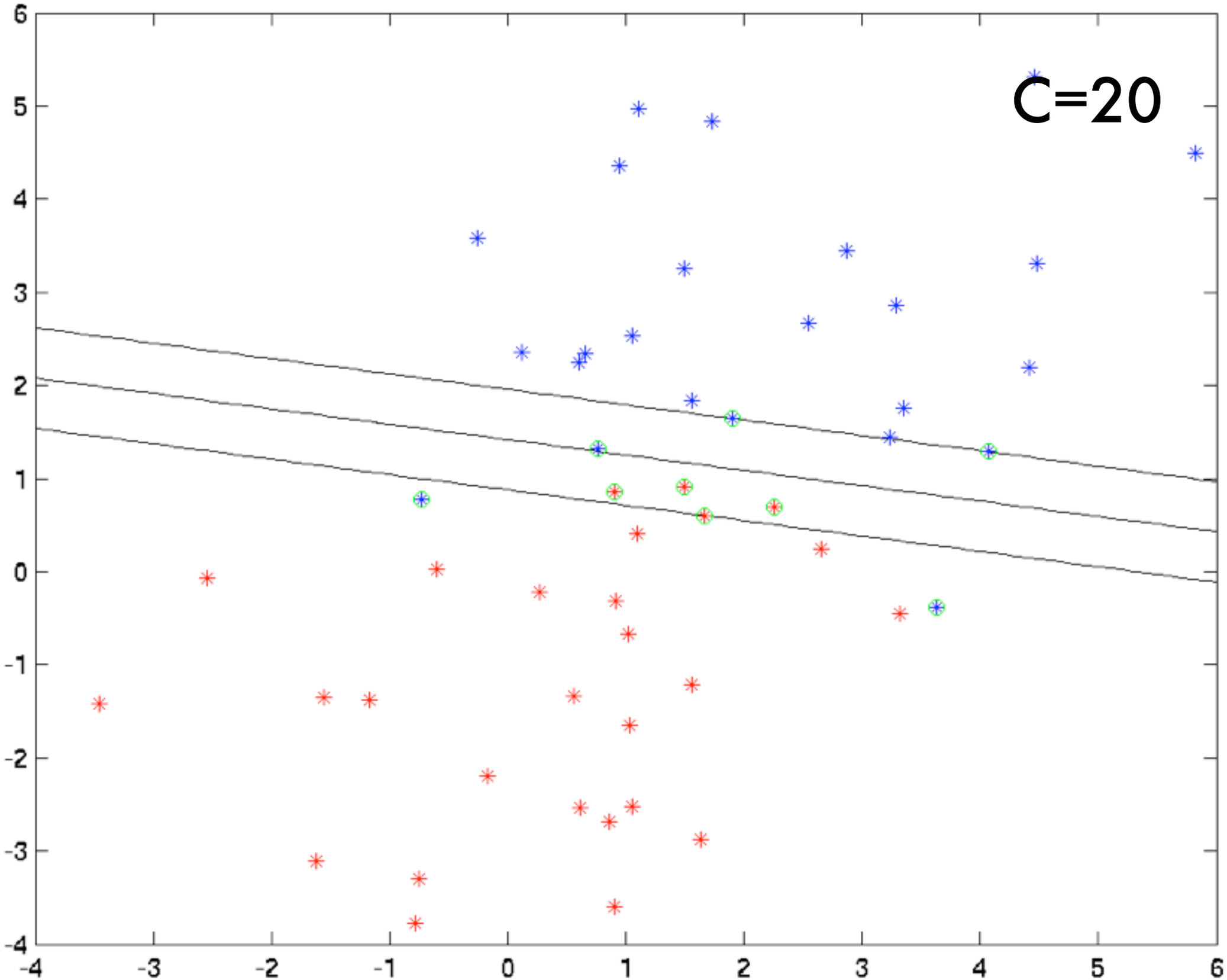

C=100
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

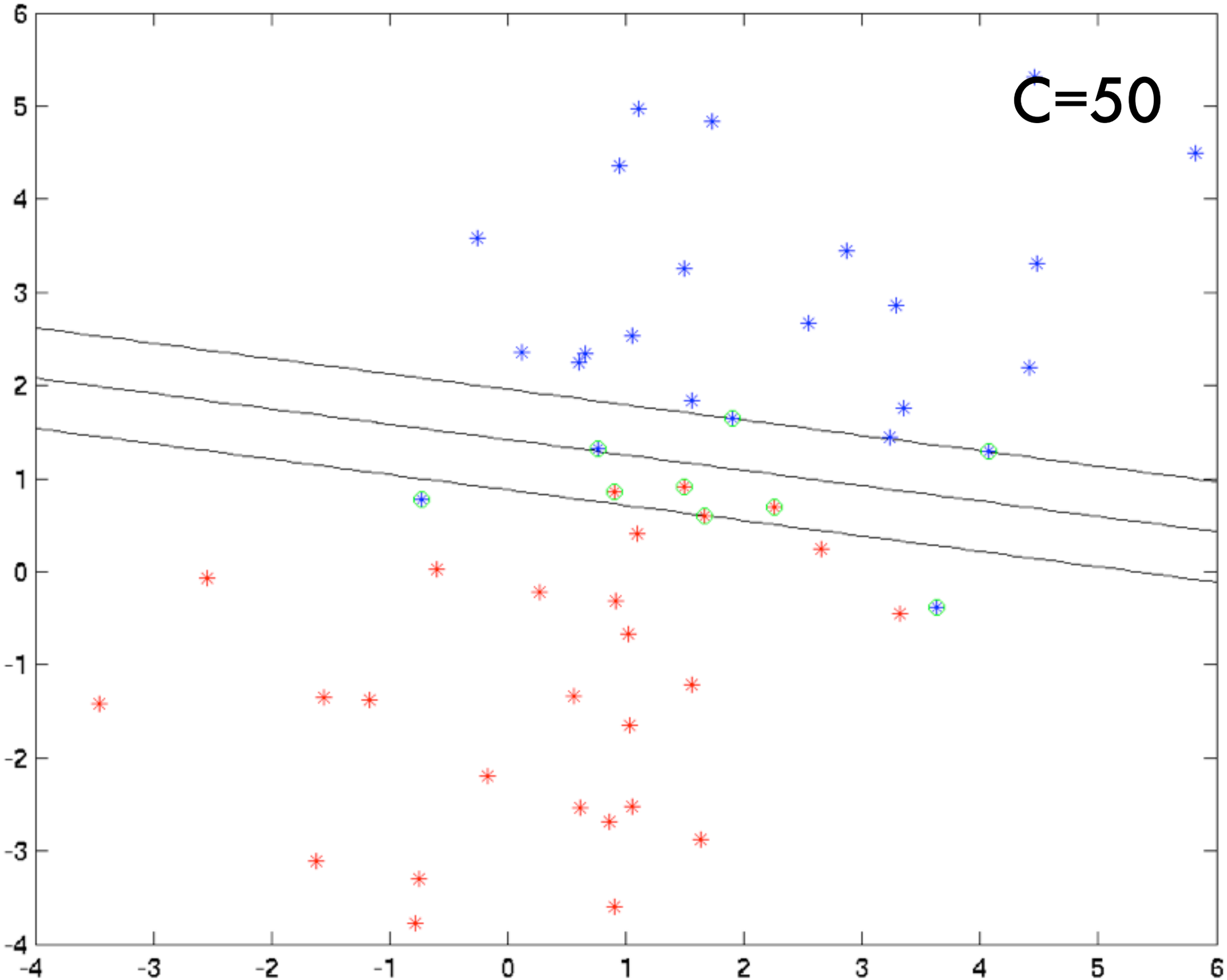

C=1
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

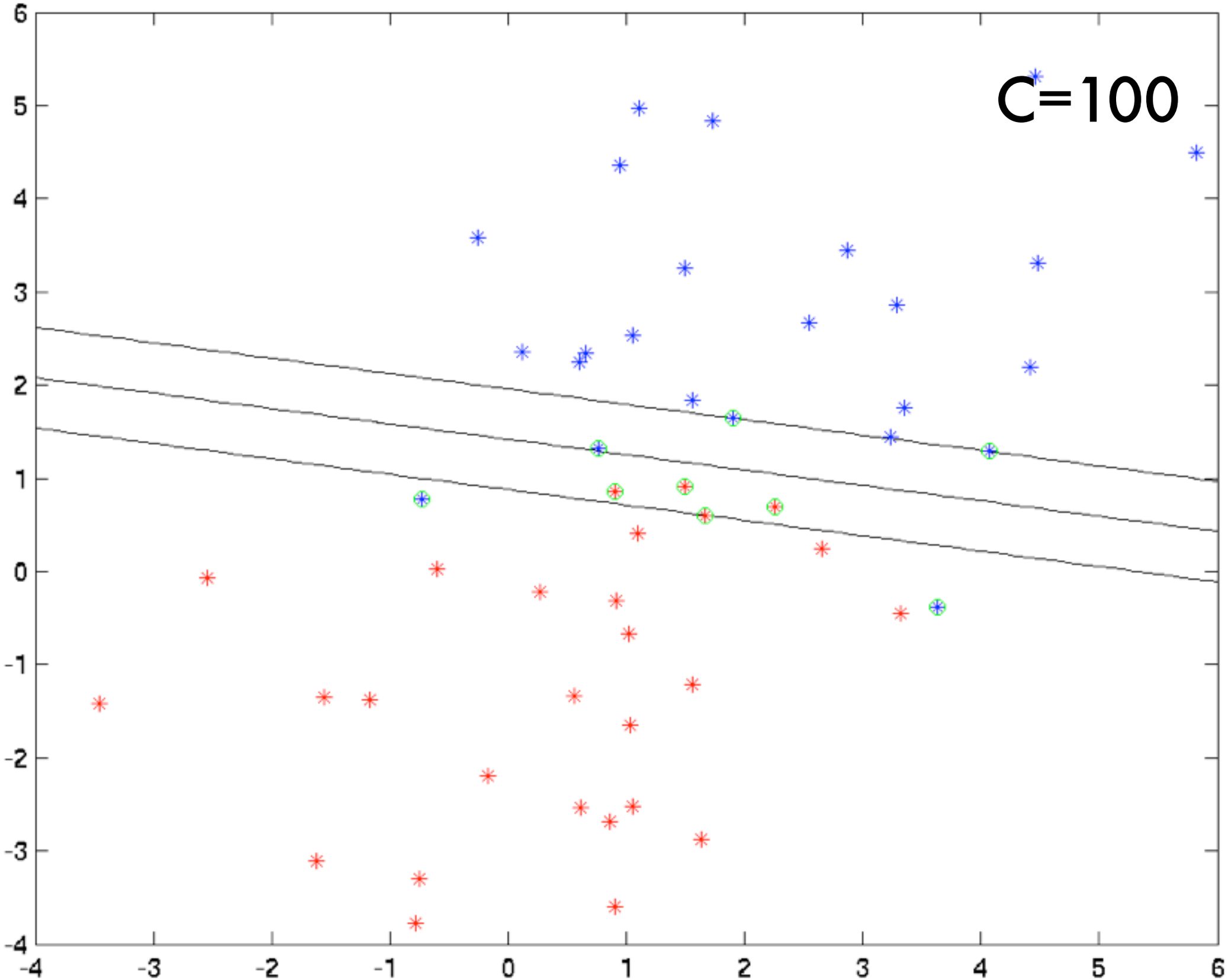

C=2
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6


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

Solving the optimization problem

- Dual problem

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

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

- 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).

Support Vector

~~Past Life~~ Regression Therapy

Certificate of Completion

This is to acknowledge that

CMU Class 10-715

has successfully completed all course requirements

This ~~25th day of March, 2012~~ 7th day of October, 2015

Authorized by: Vickie Penninger

Vickie Penninger, Reiki Master/Teacher

Risk Minimization

- Regression Loss

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

- Penalty

$$\Omega[f] \text{ e.g. } \Omega[f] = \frac{1}{2}\|w\|^2$$

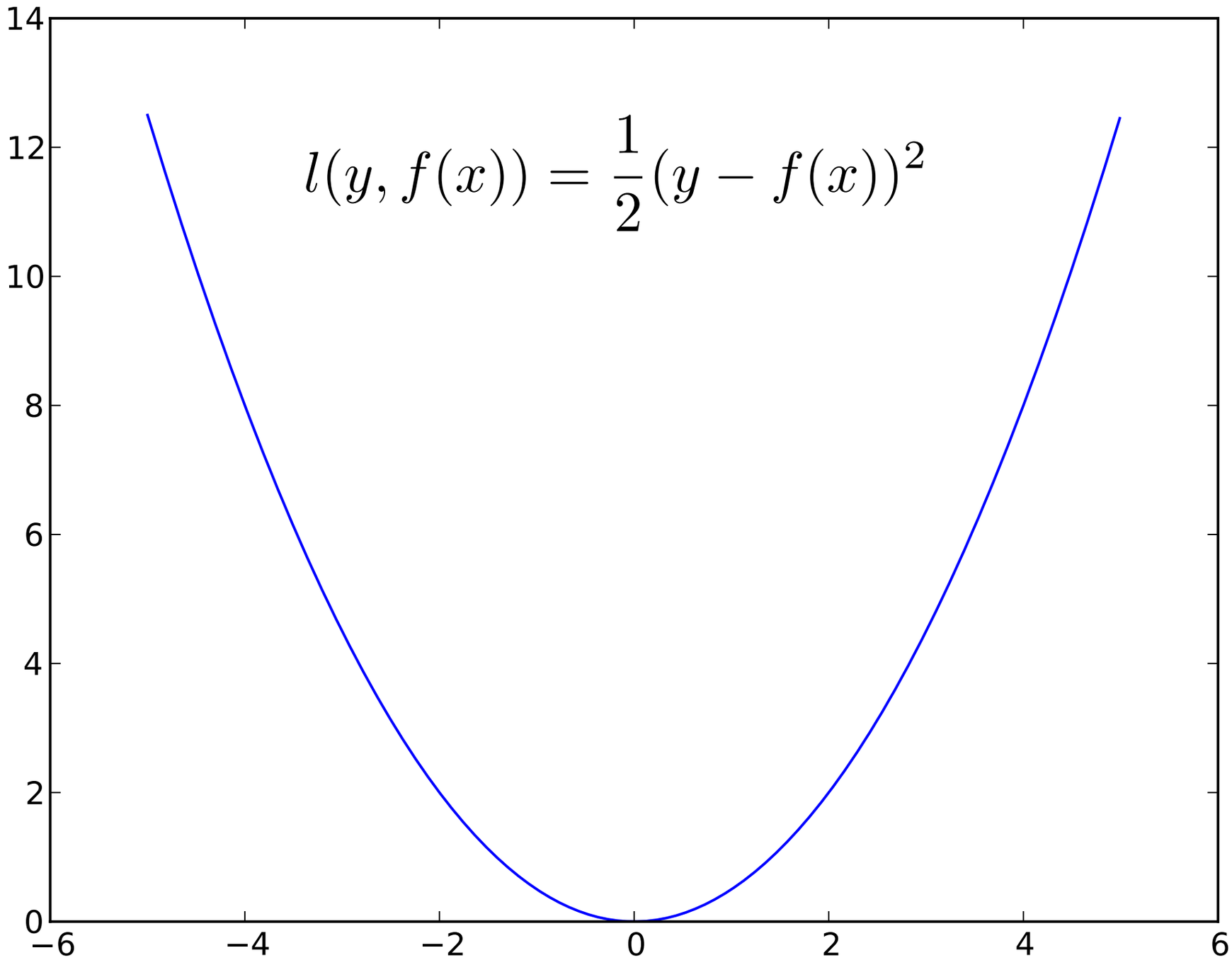
- Conversion to constrained optimization problem

$$\underset{w,\xi}{\text{minimize}} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} l(\xi_i)$$

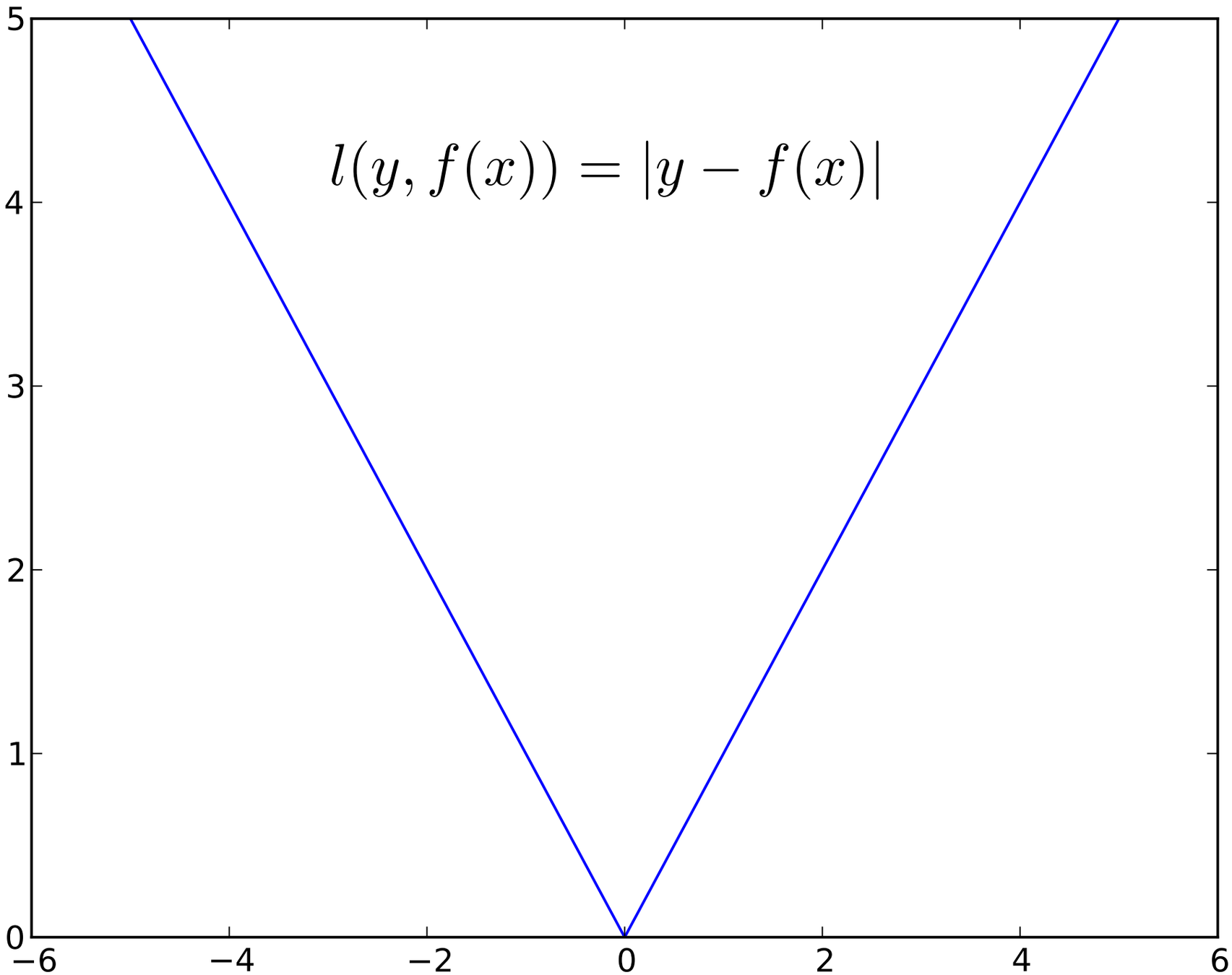
$$\text{subject to } \xi_i = y_i - [\langle w, x_i\rangle + b]$$

slack variables

Squared loss

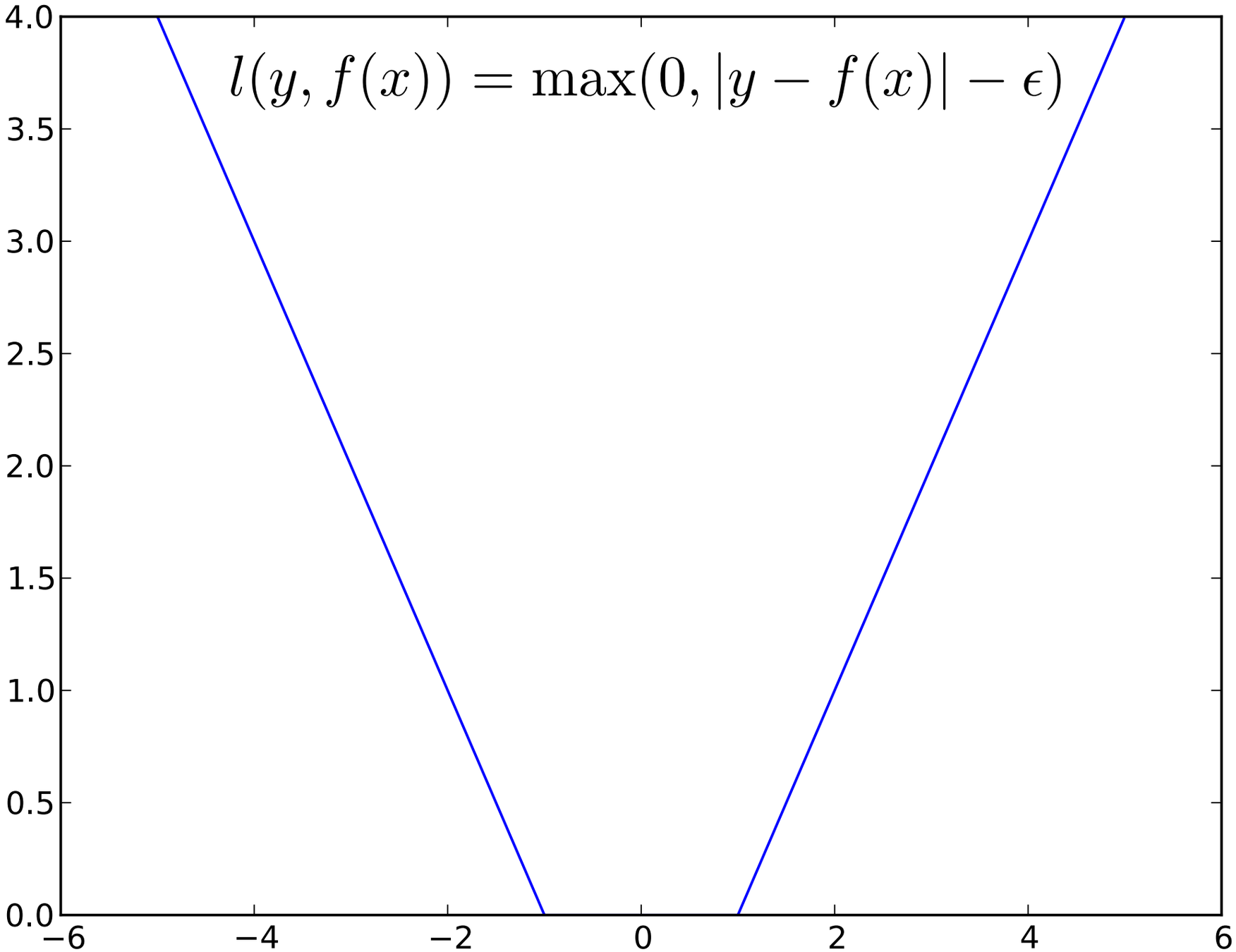


l1 loss



niversity

ε-insensitive Loss



Penalized least mean squares

- Optimization problem

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

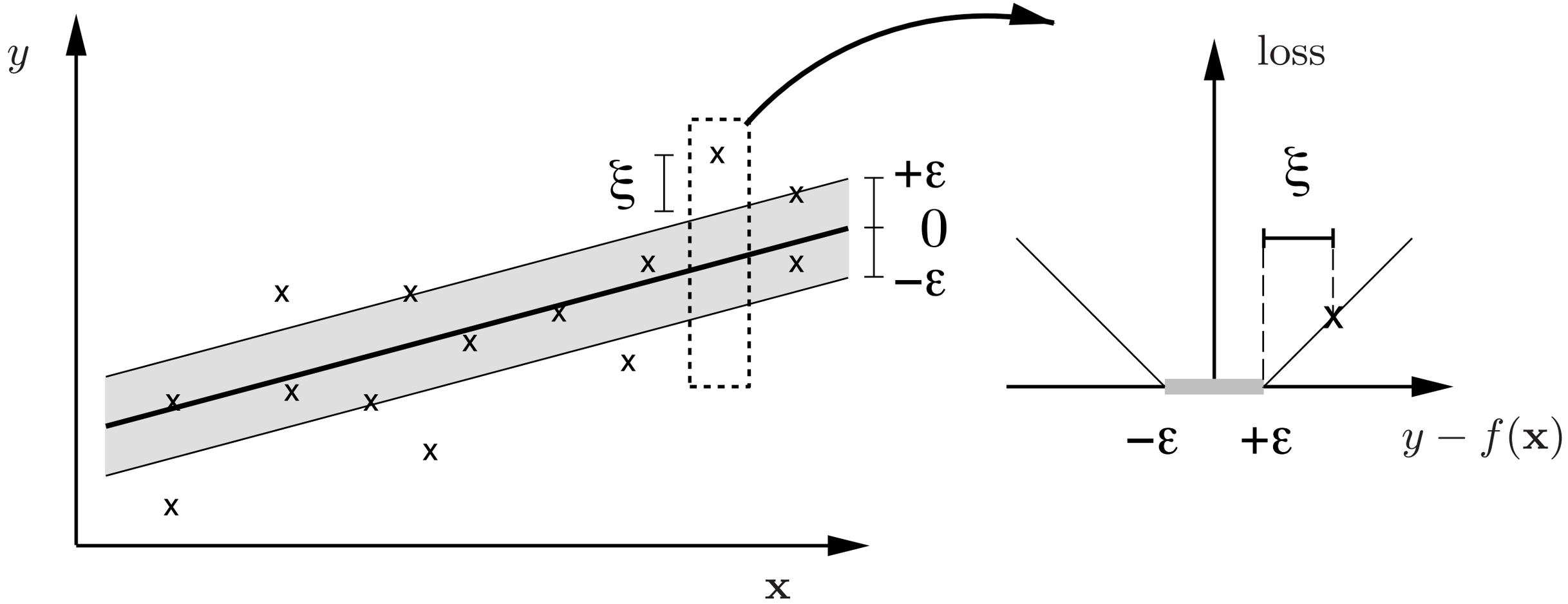
- Solution

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

Outer product matrix in X

Conjugate Gradient
Sherman Morrison Woodbury

SVM Regression



don't care about deviations within the tube

SVM Regression (ϵ-insensitive loss)

- Optimization Problem (as constrained QP)

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

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

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

- 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

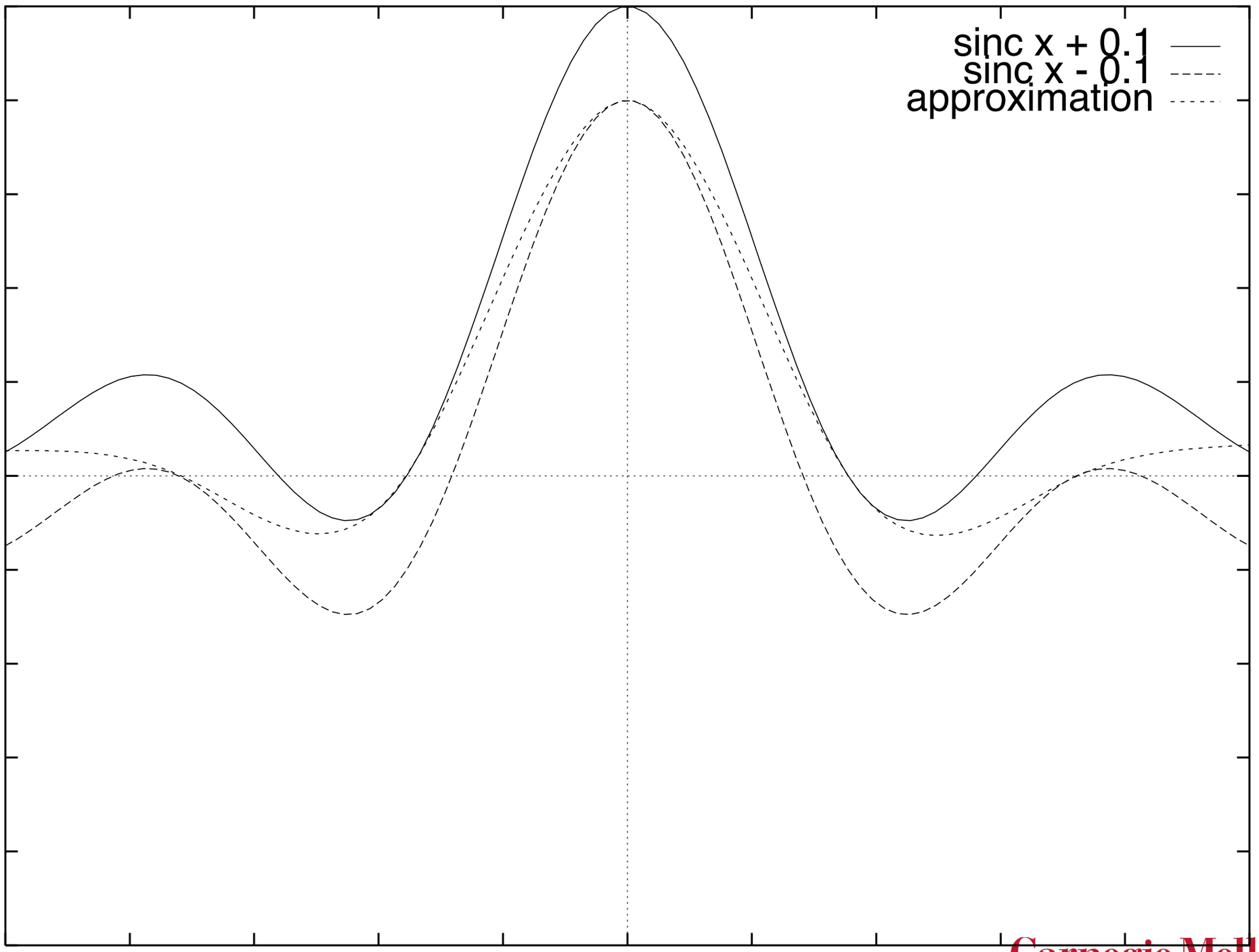
$$\underset{\alpha, \alpha^*}{\text{minimize}} \ \frac{1}{2}(\alpha - \alpha^*)^\top K(\alpha - \alpha^*) + \epsilon 1^\top (\alpha + \alpha^*) + y^\top (\alpha - \alpha^*)$$

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

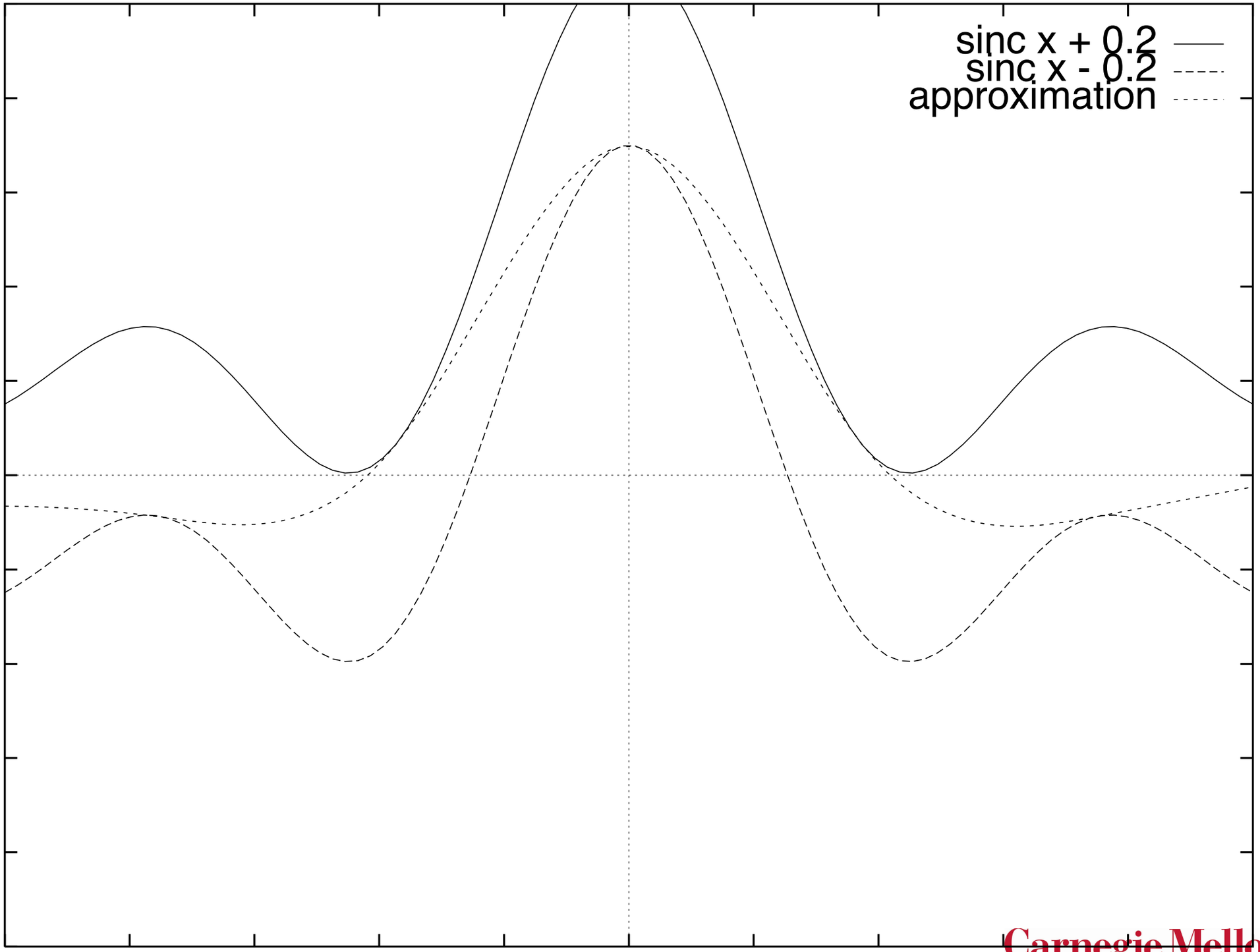
Properties

- 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

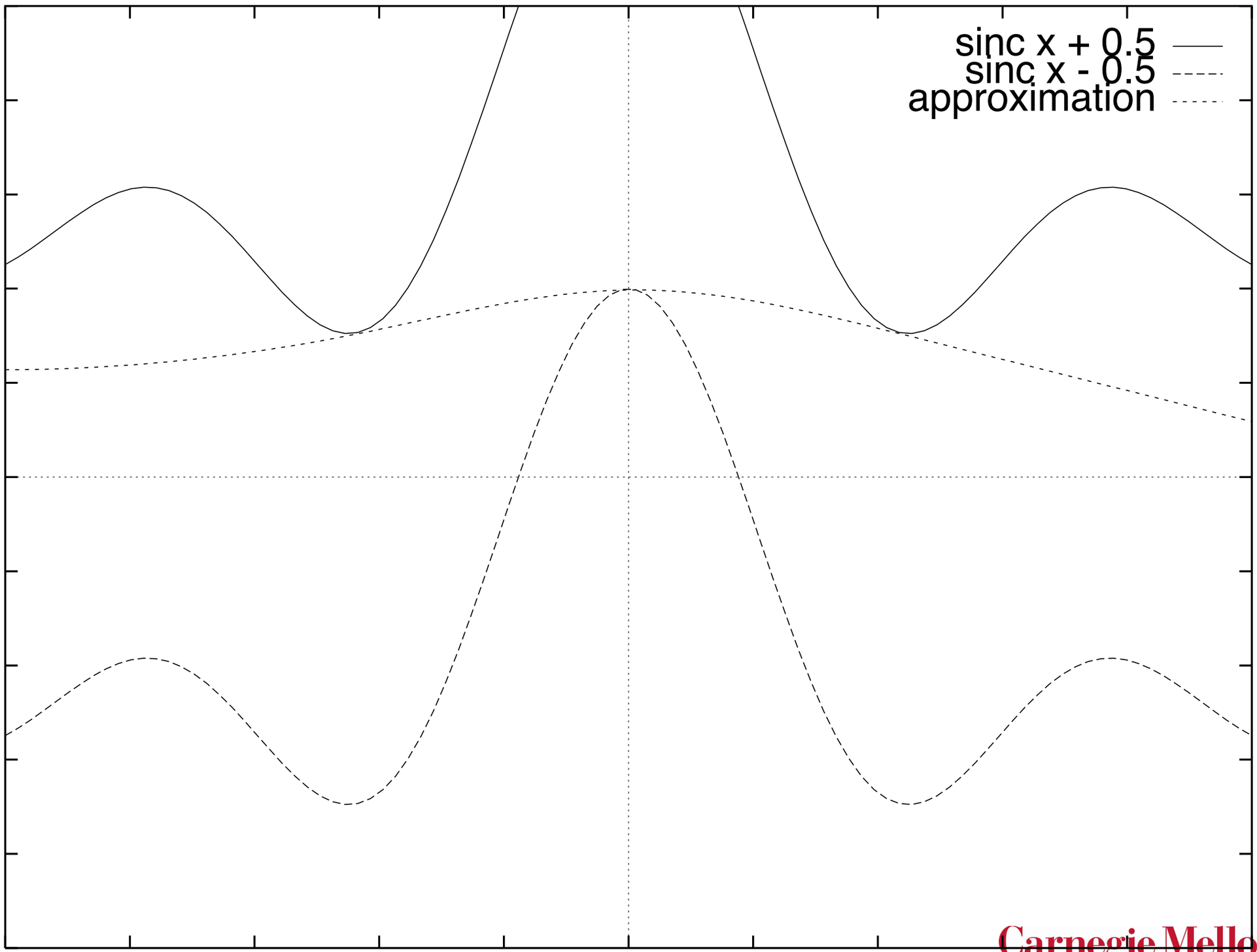
Regression example



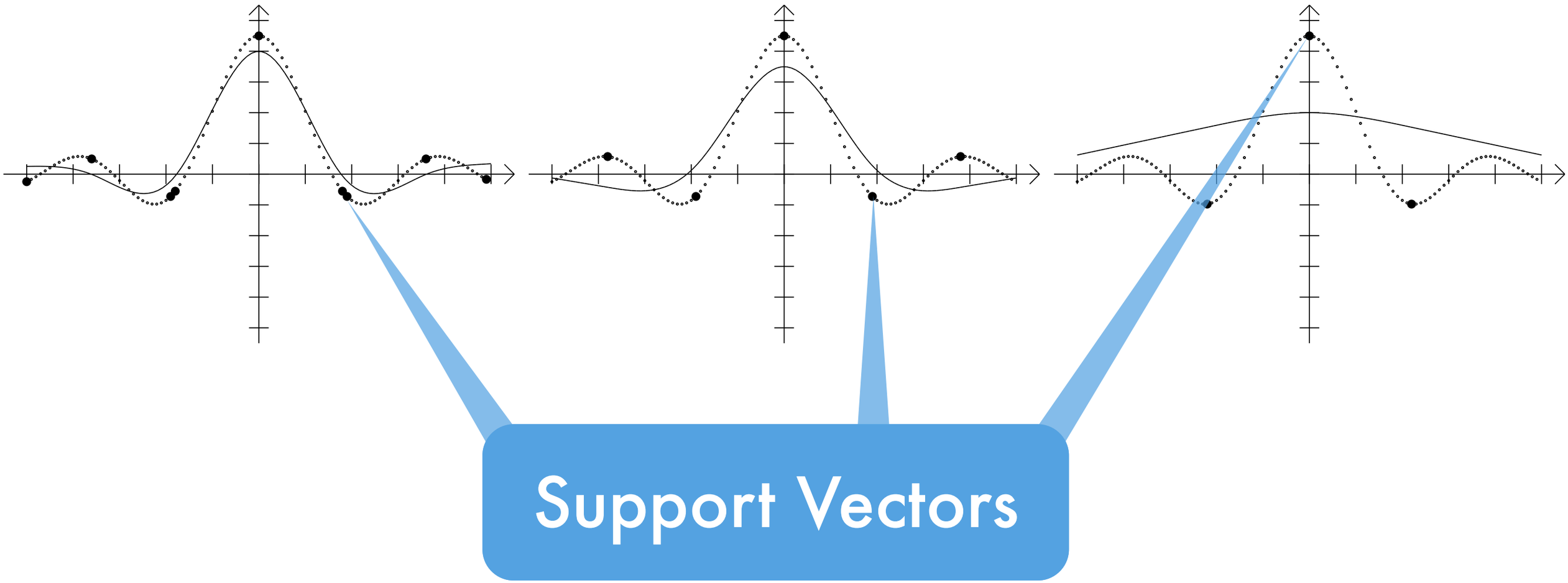
Regression example



Regression example

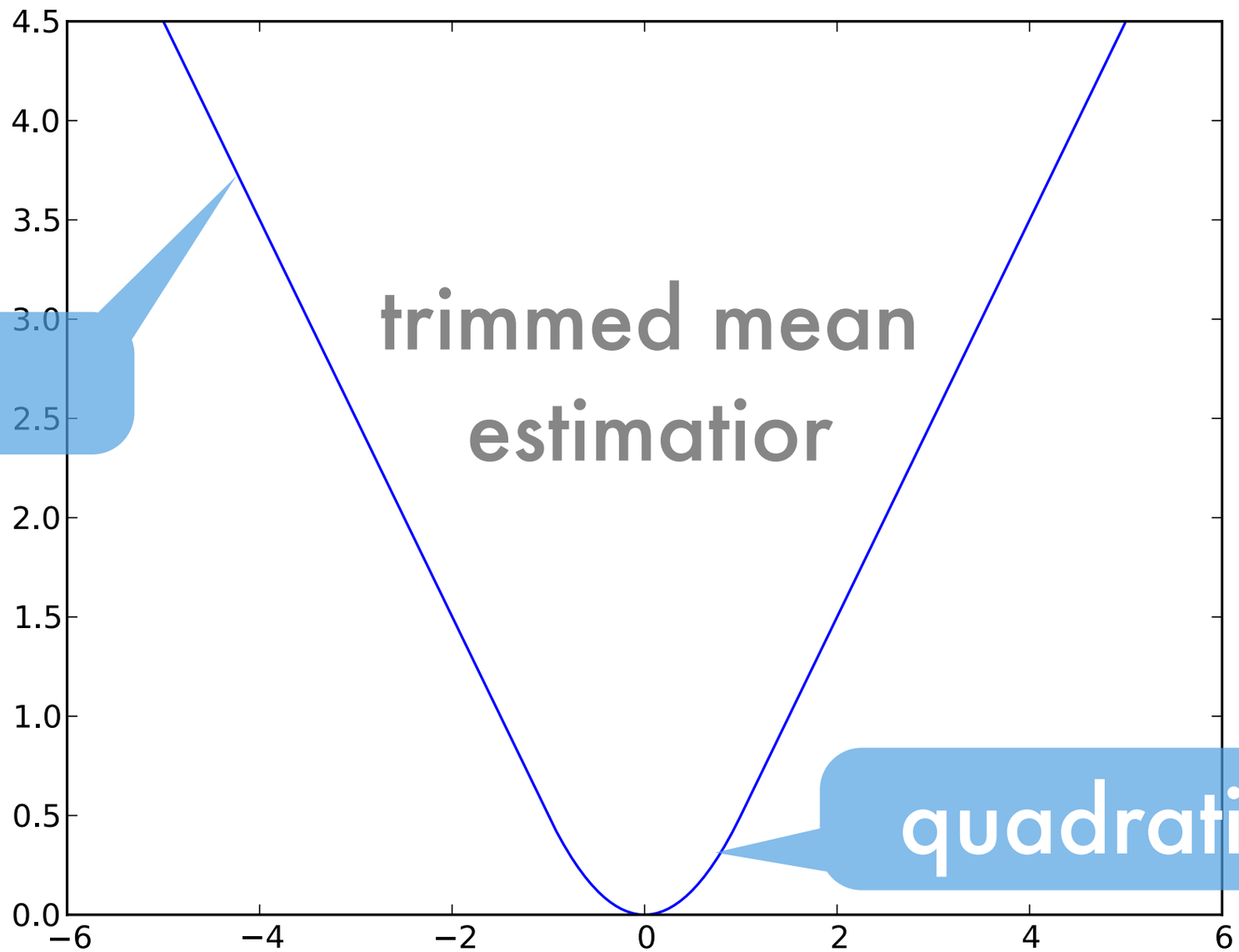


Regression example

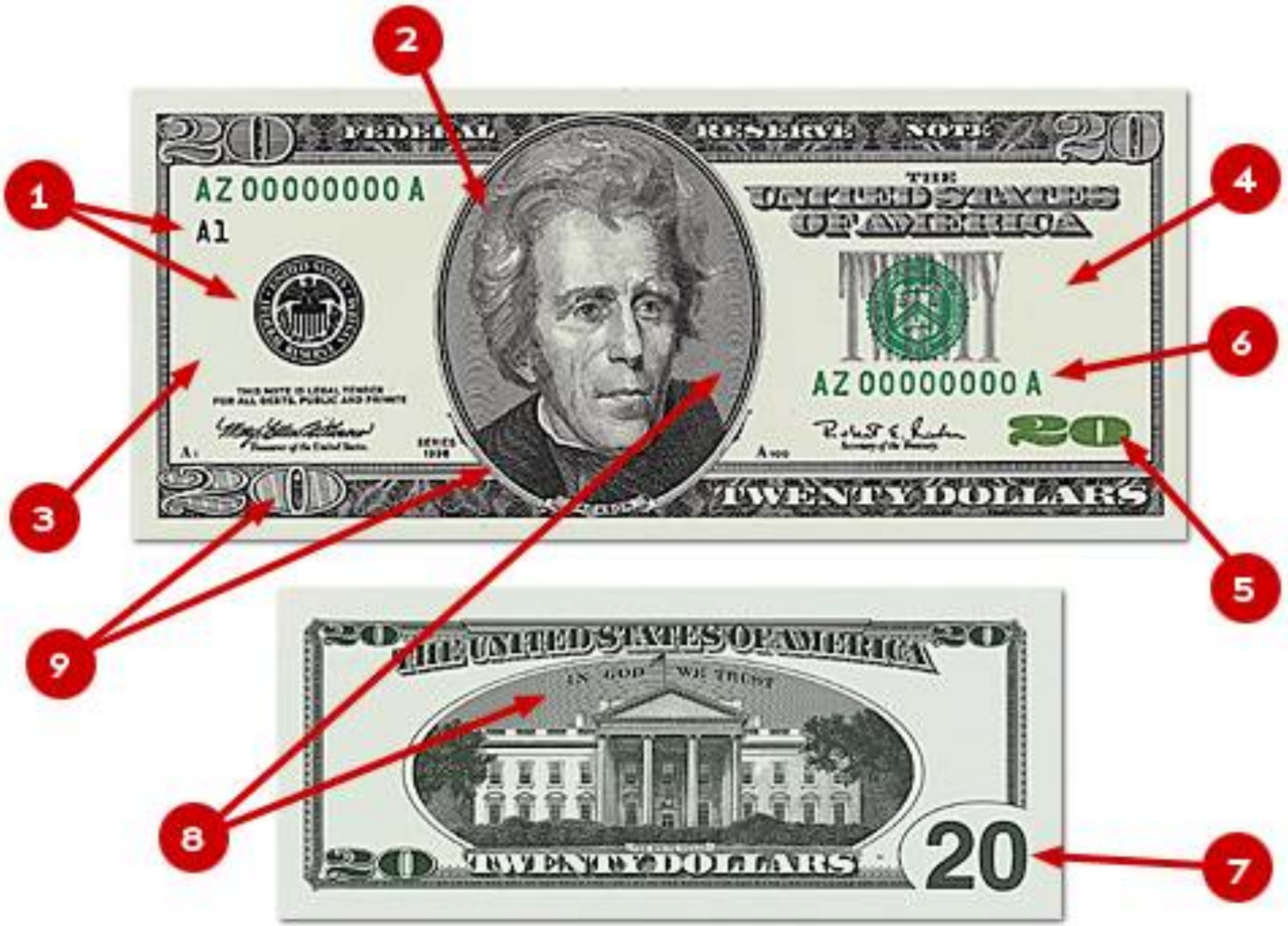


Huber's robust loss

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



Novelty Detection



Basic Idea

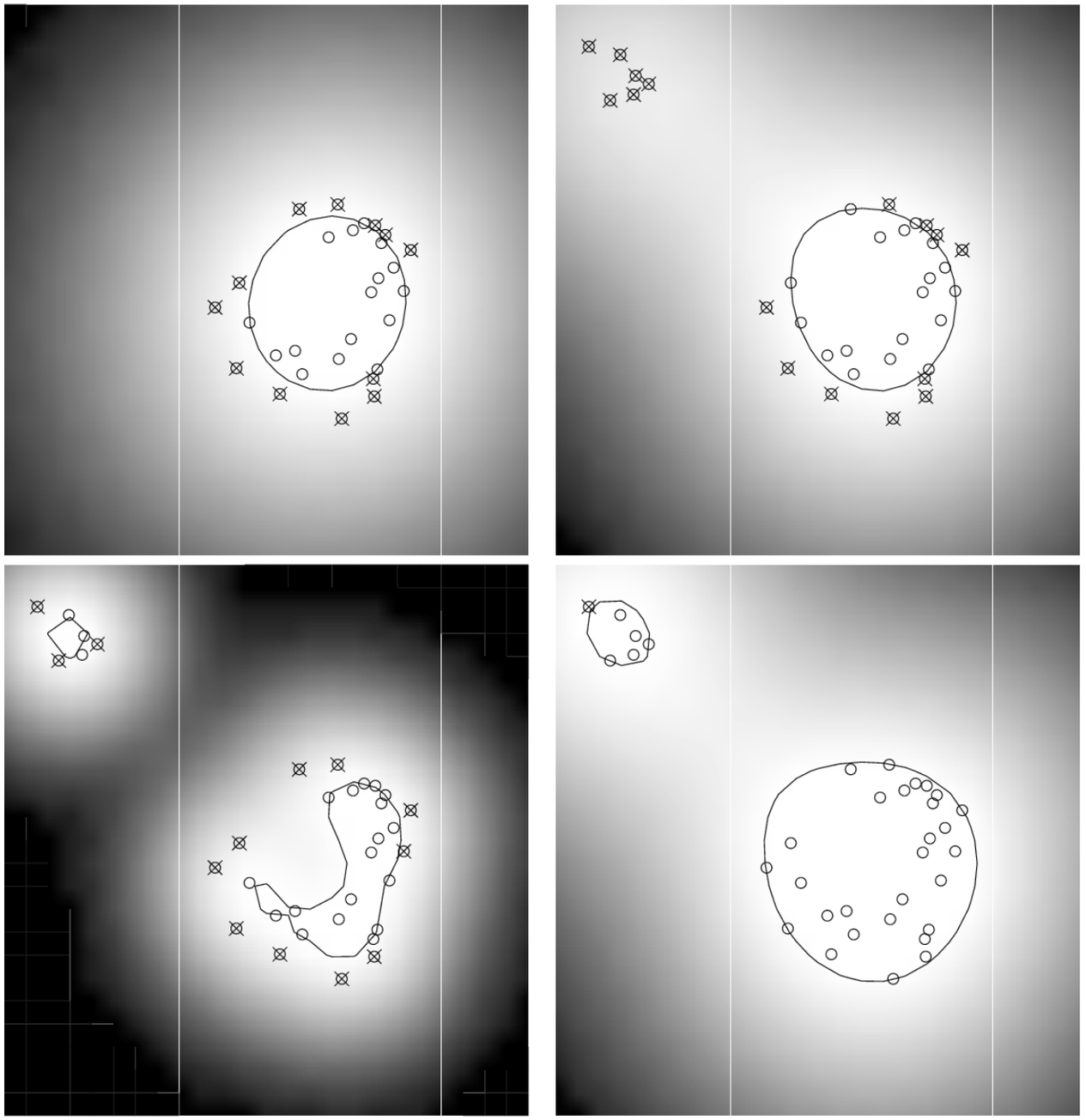
Data

Observations (x_i) generated from some $\mathrm{P}(x)$, e.g.,

- network usage patterns
- handwritten digits
- alarm sensors
- factory status

Task

Find unusual events, clean database, distinguish typical examples.



Applications

Network Intrusion Detection

Detect whether someone is trying to hack the network, downloading tons of MP3s, or doing anything else *un-usual* on the network.

Jet Engine Failure Detection

You can't destroy jet engines just to see *how* they fail.

Database Cleaning

We want to find out whether someone stored bogus information in a database (typos, etc.), mislabelled digits, ugly digits, bad photographs in an electronic album.

Fraud Detection

Credit Cards, Telephone Bills, Medical Records

Self calibrating alarm devices

Car alarms (adjusts itself to where the car is parked), home alarm (furniture, temperature, windows, etc.)

Novelty Detection via

Key Idea

- Novel data is one that we don't see frequently.
- It must lie in low density regions.

Step 1: Estimate density

- Observations $x_1, \ldots, x_m$
- Density estimate via Parzen windows

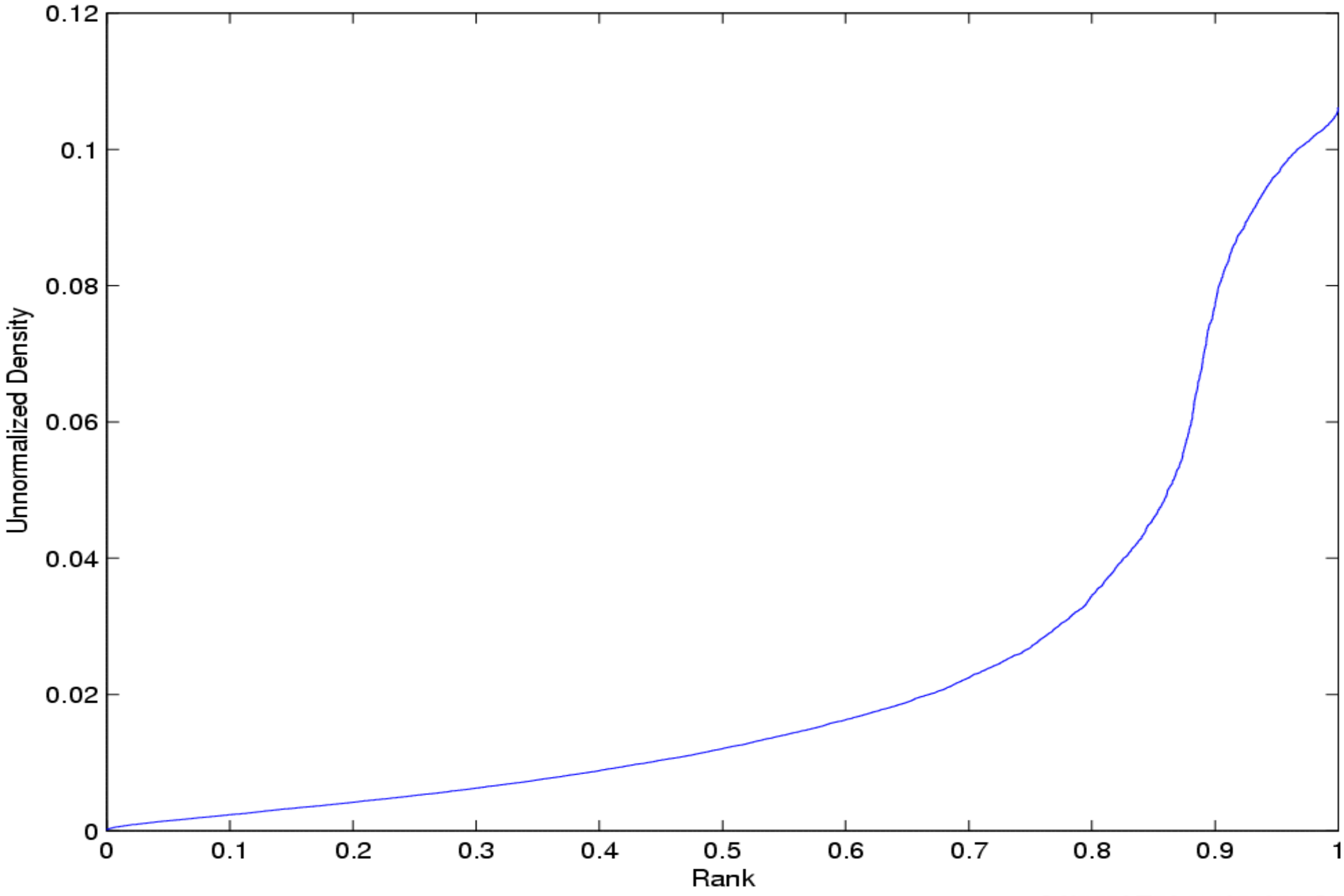
Step 2: Thresholding the density

- Sort data according to density and use it for rejection
- Practical implementation: compute

$$p(x_i) = \frac{1}{m} \sum_j k(x_i, x_j) \text{ for all } i$$

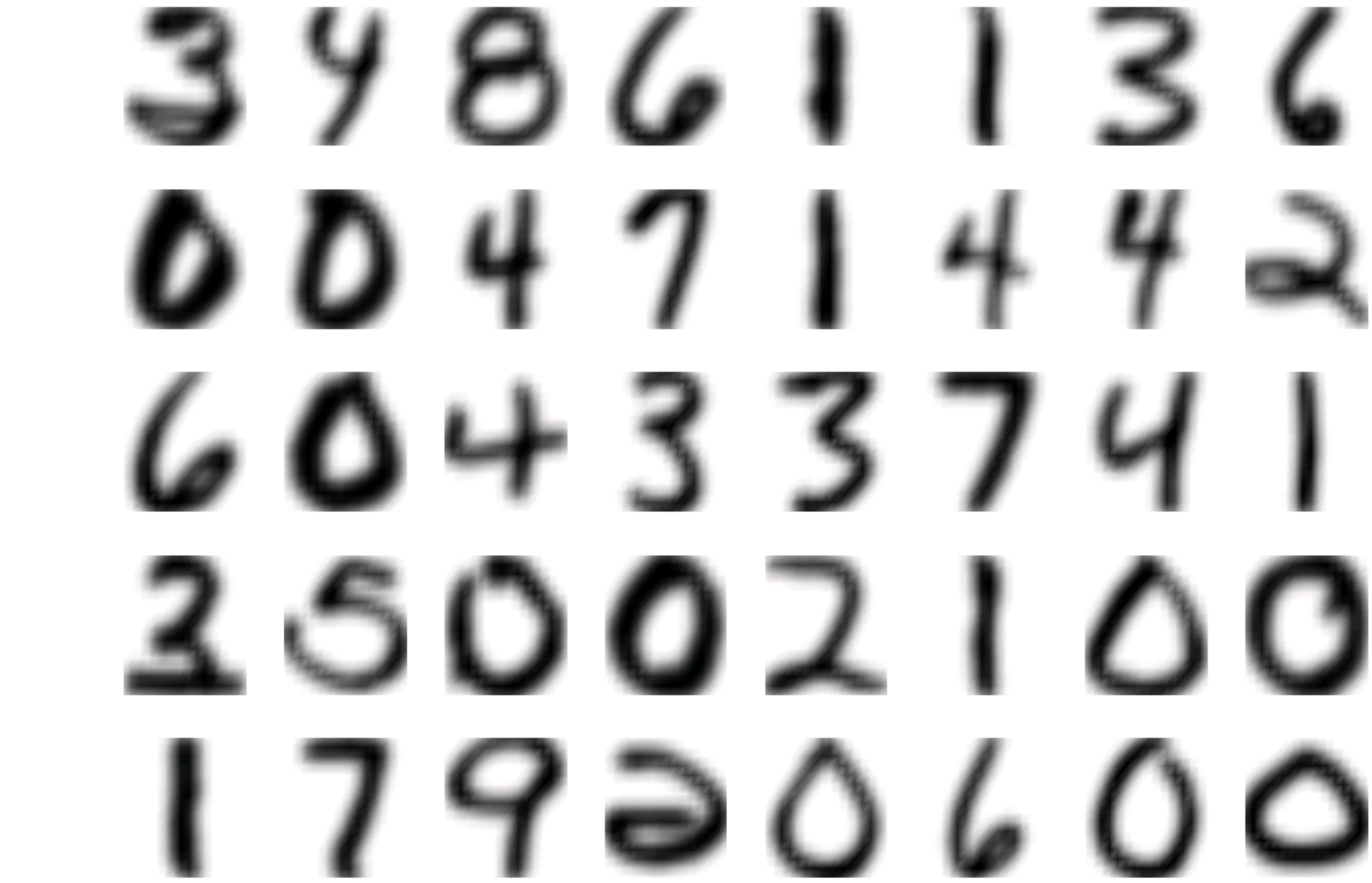
  and sort according to magnitude.
- Pick smallest $p(x_i)$ as novel points.

Order Statistics of Densities



iversity

Typical Data



Outliers

A better way

Problems

- We do not care about estimating the density properly in regions of high density (waste of capacity).
- We only care about the relative density for thresholding purposes.
- We want to eliminate a certain fraction of observations and tune our estimator specifically for this fraction.

Solution

- Areas of low density can be approximated as the **level set** of an auxiliary function. No need to estimate $p(x)$ directly — use proxy of $p(x)$.
- Specifically: find $f(x)$ such that x is novel if $f(x) \leq c$ where c is some constant, i.e. $f(x)$ describes the amount of novelty.

Problems with density estimation

- Exponential Family for density estimation

$$p(x|\theta) = \exp\left(\langle \phi(x), \theta \rangle - g(\theta)\right)$$

- MAP estimation

$$\underset{\theta}{\text{minimize}} \sum g(\theta) - \langle \phi(x_i), \theta \rangle + \frac{1}{2\sigma^2} \|\theta\|^2$$

i

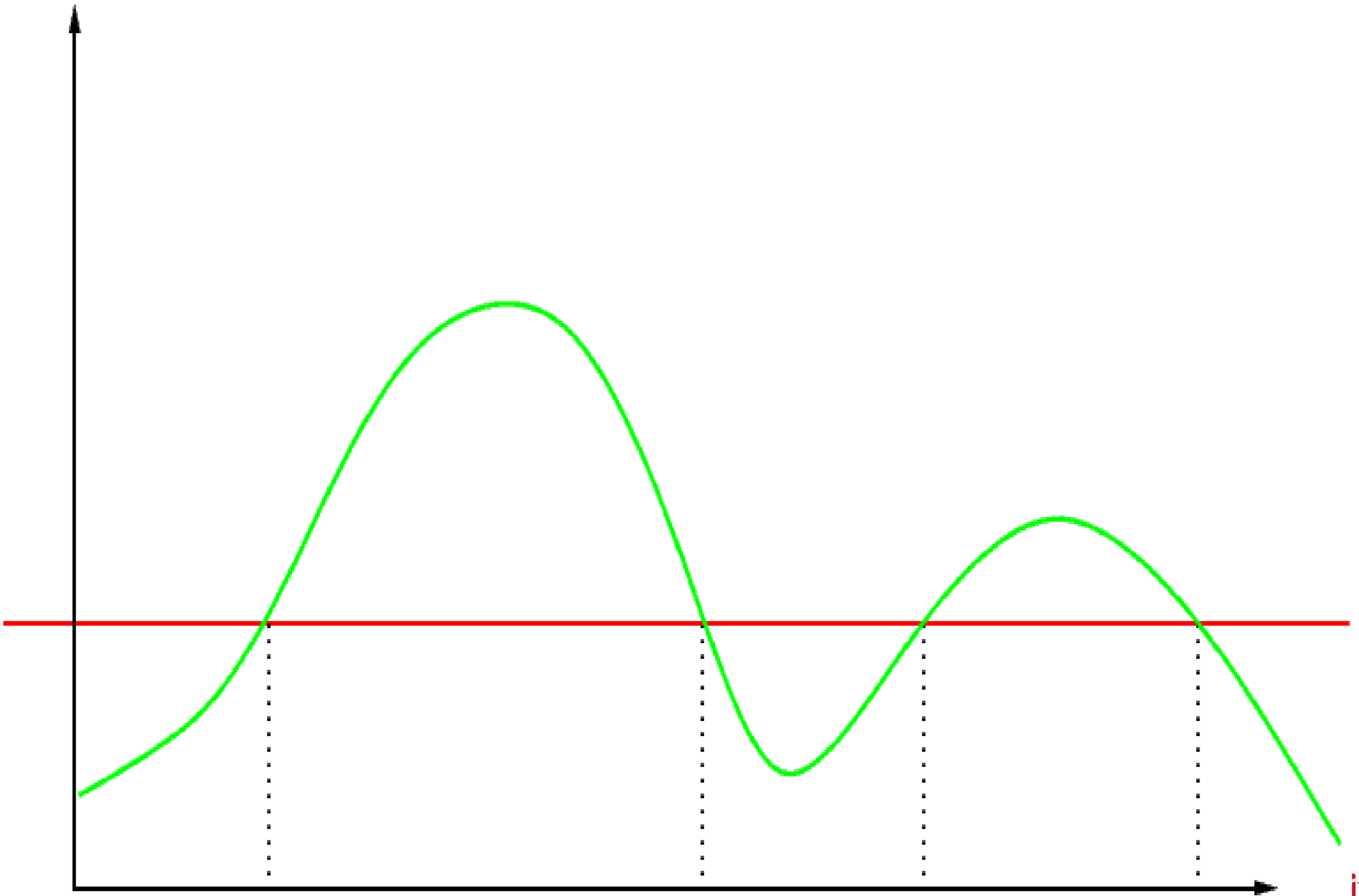
Advantages

- Convex optimization problem
- Concentration of measure

Problems

- Normalization $g(\theta)$ may be painful to compute
- For density estimation we need no normalized $p(x|\theta)$
- No need to perform particularly well in high density regions

Thresholding



Optimization Problem

Optimization Problem

$$\text{MAP} \quad \sum_{i=1}^{m} -\log p(x_i|\theta) + \frac{1}{2\sigma^2}\|\theta\|^2$$

$$\text{Novelty} \quad \sum_{i=1}^{m} \max\left(-\log \frac{p(x_i|\theta)}{\exp(\rho - g(\theta))}, 0\right) + \frac{1}{2}\|\theta\|^2$$

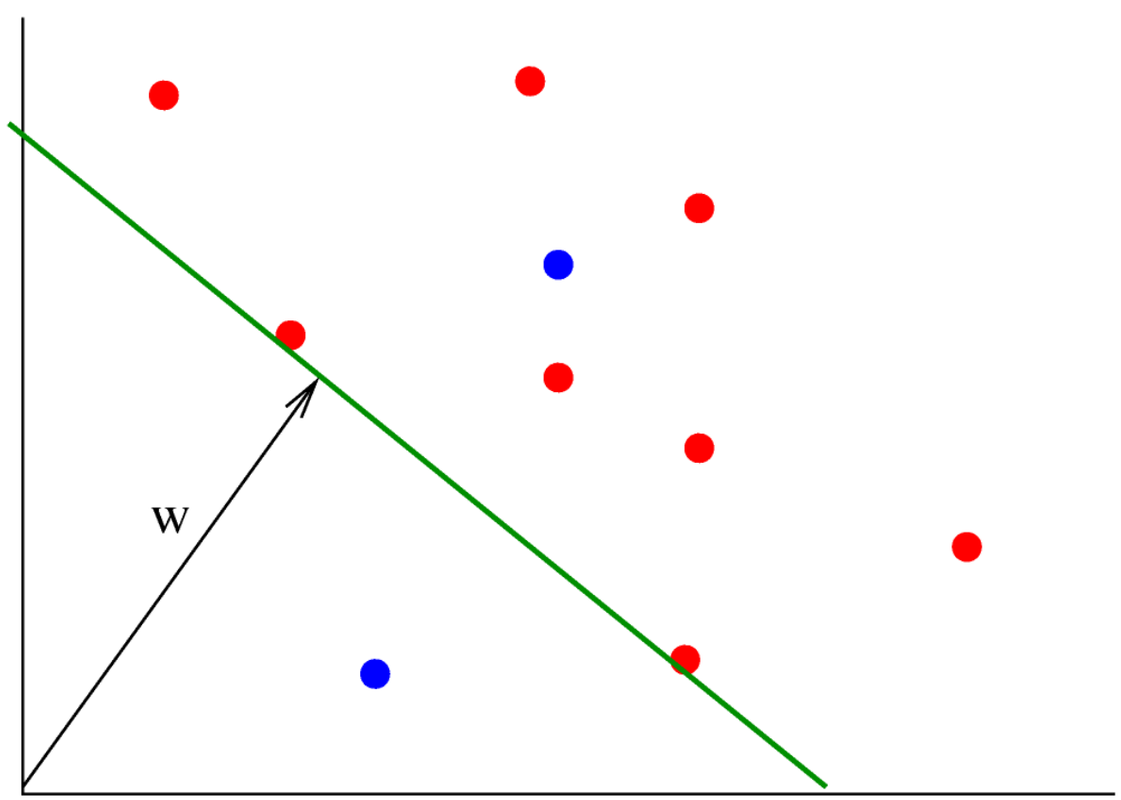
$$\sum_{i=1}^{m} \max(\rho - \langle \phi(x_i), \theta \rangle, 0) + \frac{1}{2}\|\theta\|^2$$

Advantages

- No normalization $g(\theta)$ needed
- No need to perform particularly well in high density regions (estimator focuses on low-density regions)
- Quadratic program

Maximum Distance

Idea Find hyperplane, given by $f(x) = \langle w, x \rangle + b = 0$ that has **maximum distance from origin** yet is still closer to the origin than the observations.



Hard Margin

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|w\|^2 \\ \text{subject to} \quad & \langle w, x_i \rangle \geq 1 \end{aligned}$$

Soft Margin

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{m} \xi_i \\ \text{subject to} \quad & \langle w, x_i \rangle \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned}$$

Optimization Problem

Primal Problem

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$\text{subject to} \quad \langle w, x_i\rangle - 1 + \xi_i \geq 0 \text{ and } \xi_i \geq 0$$

Lagrange Function L

- Subtract constraints, multiplied by Lagrange multipliers (α_i and η_i), from Primal Objective Function.
- Lagrange function L has **saddlepoint** at optimum.

$$L = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left(\langle w, x_i\rangle - 1 + \xi_i\right) - \sum_{i=1}^{m}\eta_i\xi_i$$

subject to $\alpha_i, \eta_i \geq 0$.

Dual Problem

Optimality Conditions

$$\partial_w L = w - \sum_{i=1}^{m} \alpha_i x_i = 0 \implies w = \sum_{i=1}^{m} \alpha_i x_i$$

$$\partial_{\xi_i} L = C - \alpha_i - \eta_i = 0 \implies \alpha_i \in [0, C]$$

Now **substitute** the optimality conditions **back into** L.

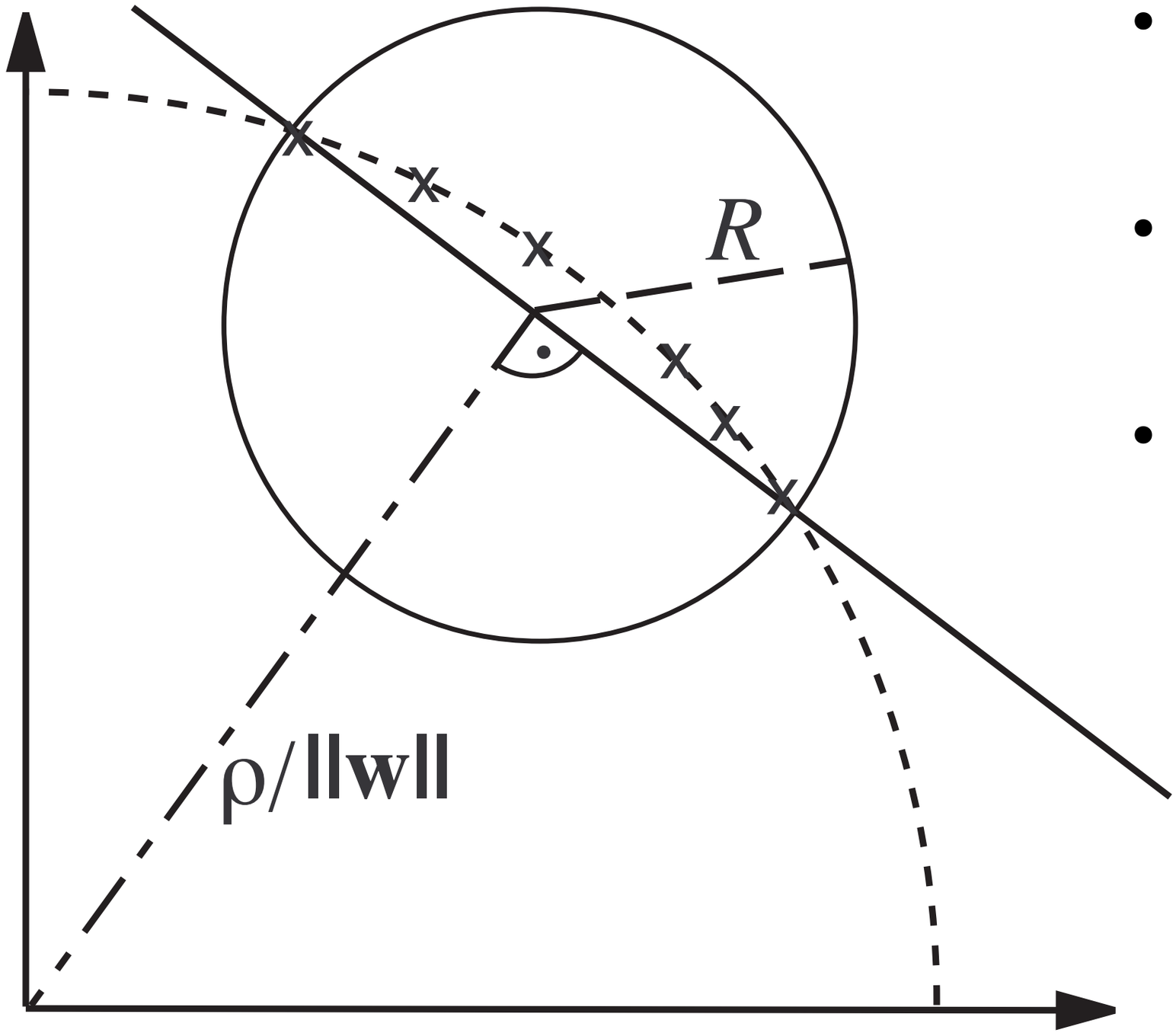
Dual Problem

$$\text{minimize} \quad \frac{1}{2} \sum_{i=1}^{m} \alpha_i \alpha_j \langle x_i, x_j \rangle - \sum_{i=1}^{m} \alpha_i$$

$$\text{subject to} \quad \alpha_i \in [0, C]$$

All this is only possible due to the convexity of the primal problem.

Minimum enclosing ball



- Observations on surface of ball
- Find minimum enclosing ball
- Equivalent to single class SVM

Adaptive thresholds

Problem

- Depending on C, the number of novel points will vary.
- We would like to **specify the fraction** ν beforehand.

Solution

Use hyperplane separating data from the origin

$$H := \{x | \langle w, x \rangle = \rho\}$$

where the threshold ρ is **adaptive**.

Intuition

- Let the hyperplane shift by shifting ρ
- Adjust it such that the 'right' number of observations is considered novel.
- Do this automatically

Optimization Problem

Primal Problem

$$\text{minimize } \frac{1}{2}\|w\|^2 + \sum_{i=1}^{m} \xi_i - m\nu\rho$$

$$\text{where } \langle w, x_i \rangle - \rho + \xi_i \geq 0$$

$$\xi_i \geq 0$$

Dual Problem

$$\text{minimize } \frac{1}{2} \sum_{i=1}^{m} \alpha_i \alpha_j \langle x_i, x_j \rangle$$

$$\text{where } \alpha_i \in [0, 1] \text{ and } \sum_{i=1}^{m} \alpha_i = \nu m.$$

The v-property theorem

- Optimization problem

$$\underset{w}{\text{minimize}} \ \frac{1}{2}\|w\|^2 + \sum_{i=1}^{m} \xi_i - m\nu\rho$$

$$\text{subject to } \langle w, x_i \rangle \geq \rho - \xi_i \text{ and } \xi_i \geq 0$$

- Solution satisfies
  - At most a fraction of v points are novel
  - At most a fraction of (1-v) points aren't novel
  - Fraction of points on boundary vanishes for large m (for non-pathological kernels)

Proof

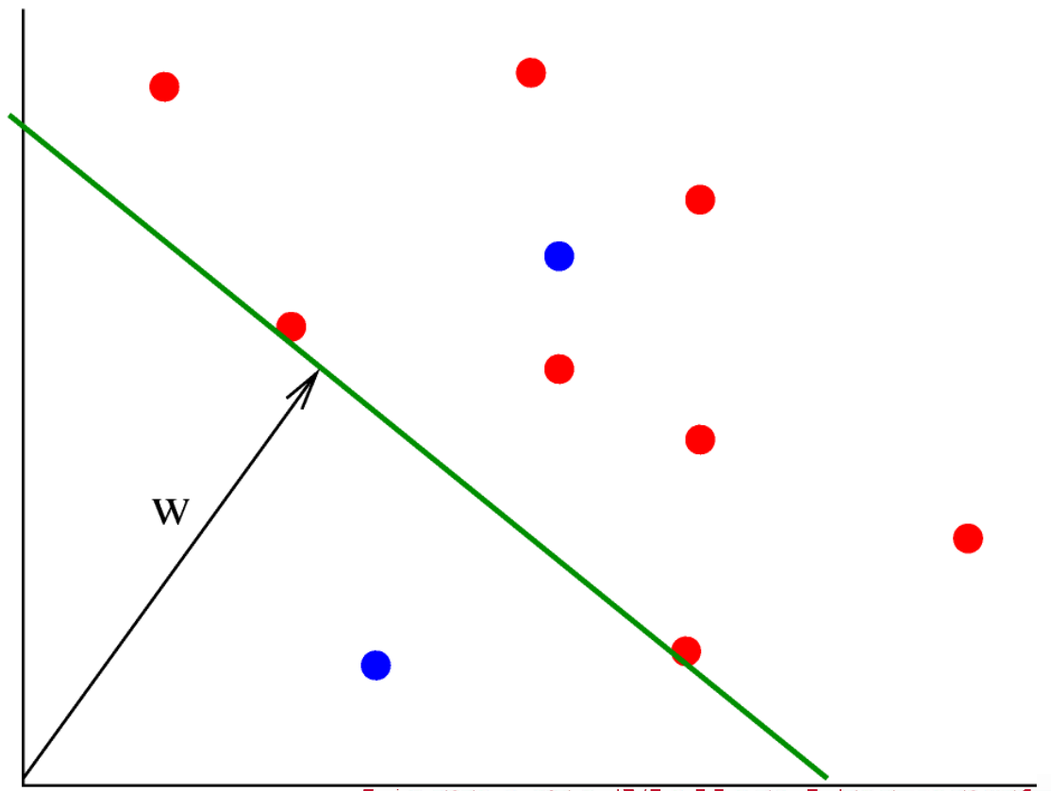
- Move boundary at optimality
  - For smaller threshold m- points on wrong side of margin contribute $\delta(m_- - \nu m) \leq 0$
  - For larger threshold m+ points not on ‘good’ side of margin yield

$$\delta(m_+ - \nu m) \geq 0$$

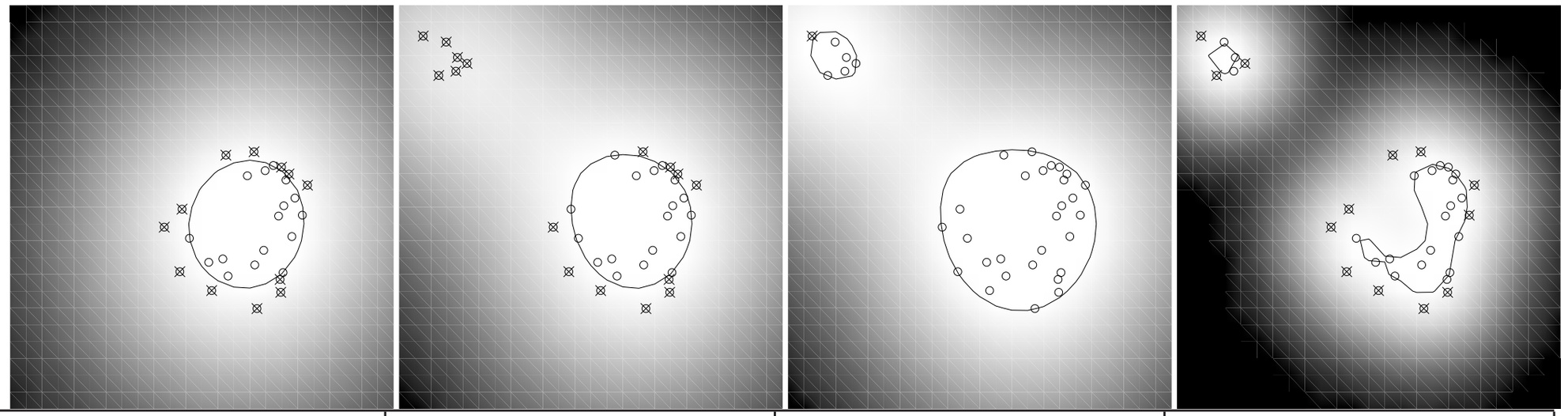
  - Combining inequalities

$$\frac{m_-}{m} \leq \nu \leq \frac{m_+}{m}$$

- Margin set of measure 0



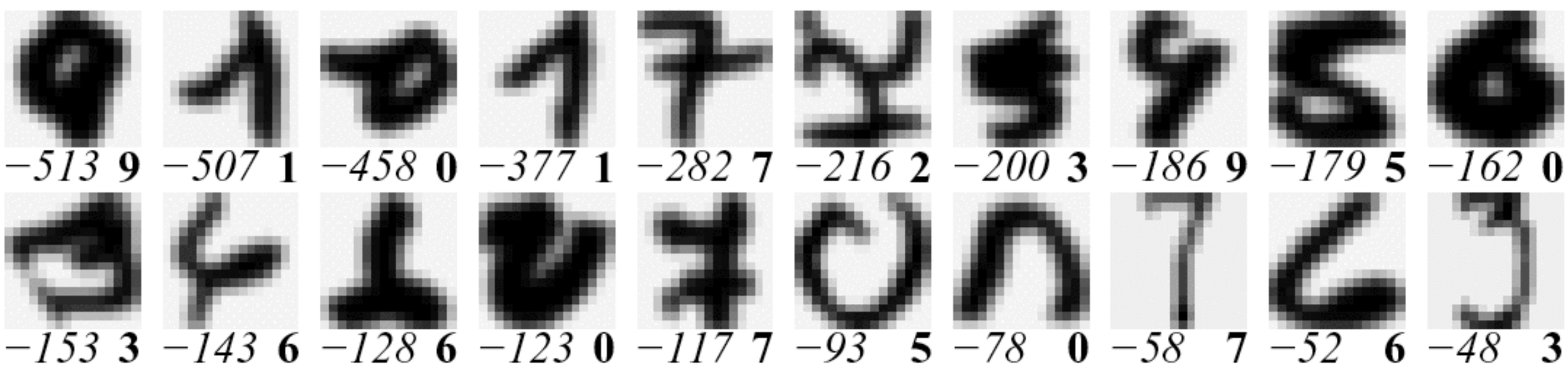
Toy example



ν, width c	0.5, 0.5	0.5, 0.5	0.1, 0.5	0.5, 0.1
frac. SVs/OLs	0.54, 0.43	0.59, 0.47	0.24, 0.03	0.65, 0.38
margin $\rho/\|\mathbf{w}\|$	0.84	0.70	0.62	0.48

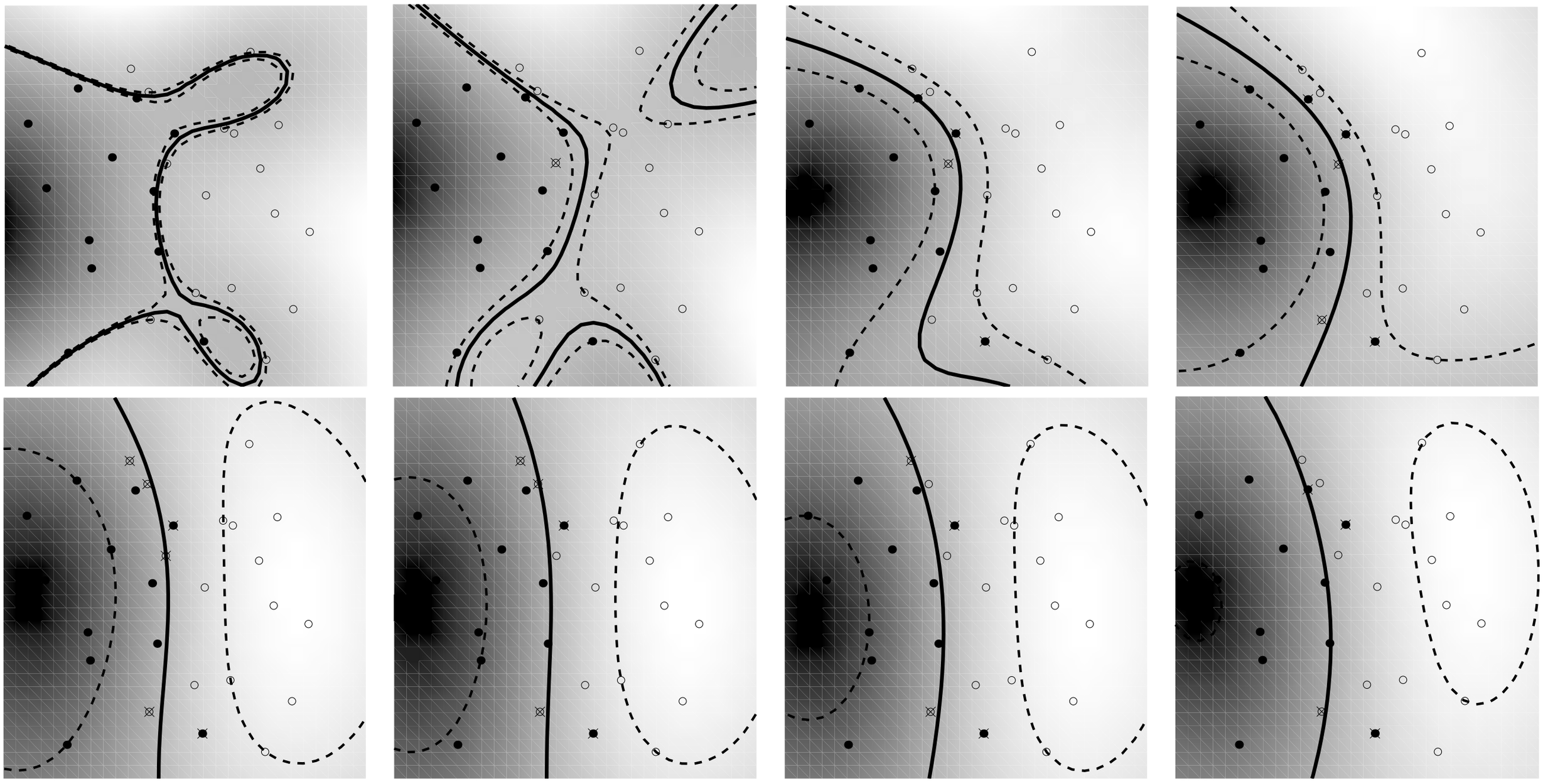
threshold and smoothness requirements

Novelty detection for OCR



- Better estimates since we only optimize in low density regions.
- Specifically tuned for small number of outliers.
- Only estimates of a level-set.
- For $\nu = 1$ we get the Parzen-windows estimator back.

Classification with the v-

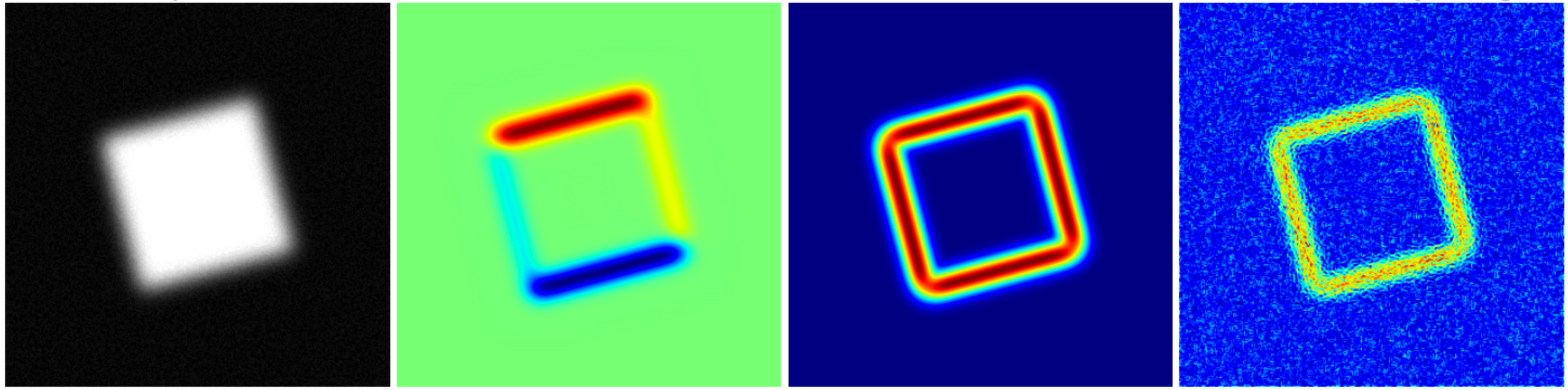


changing kernel width and threshold

Outline

- **Convex Optimization**
  - Unconstrained Optimization
  - Constrained Optimization and Duality
  - Linear and Quadratic programs
- **Support Vector Machines**
  - Classification
  - Regression
  - Novelty Detection
- **Kernels**
  - Feature Space
  - Kernel PCA
  - Kernelized SVM

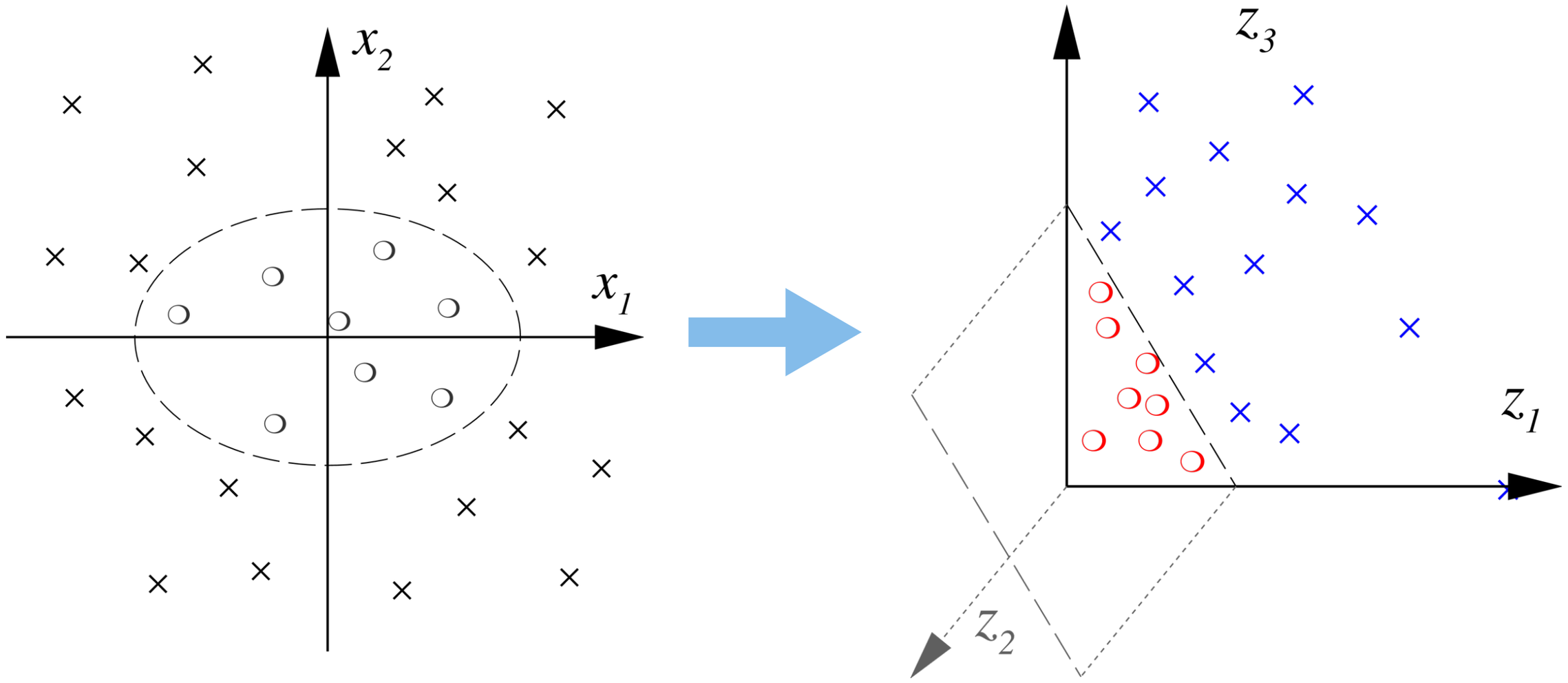
Preprocessing



Nonlinear Features

- Regression
  We got nonlinear functions by preprocessing
- Perceptron
  - Map data into feature space $x \to \phi(x)$
  - Solve problem in this space
  - Query replace $\langle x, x' \rangle$ by $\langle \phi(x), \phi(x') \rangle$ for code
- Feature Perceptron
  - Solution in span of $\phi(x_i)$

Quadratic Features



- Separating surfaces are Circles, hyperbolae, parabolae

Constructing Features

	1	2	3	4	5	6	7	8	9	0
Loops	0	0	0	1	0	1	0	2	1	1
3 Joints	0	0	0	0	0	1	0	0	1	0
4 Joints	0	0	0	1	0	0	0	1	0	0
Angles	0	1	1	1	1	0	1	0	0	0
Ink	1	2	2	2	2	2	1	3	2	2

```
Delivered-To: alex.smola@gmail.com
Received: by 10.216.47.73 with SMTP id s51cs361171web;
        Tue, 3 Jan 2012 14:17:53 -0800 (PST)
Received: by 10.213.17.145 with SMTP id s17mr2519891eba.147.1325629071725;
        Tue, 03 Jan 2012 14:17:51 -0800 (PST)
Return-Path: <alex+caf_=alex.smola=gmail.com@smola.org>
Received: from mail-ey0-f175.google.com (mail-ey0-f175.google.com [209.85.215.175])
        by mx.google.com with ESMTPS id n4si29264232eef.57.2012.01.03.14.17.51
        (version=TLSv1/SSLv3 cipher=OTHER);
        Tue, 03 Jan 2012 14:17:51 -0800 (PST)
Received-SPF: neutral (google.com: 209.85.215.175 is neither permitted nor denied by best
guess record for domain of alex+caf_=alex.smola=gmail.com@smola.org) client-
ip=209.85.215.175;
Authentication-Results: mx.google.com; spf=neutral (google.com: 209.85.215.175 is neither
permitted nor denied by best guess record for domain of alex
+caf_=alex.smola=gmail.com@smola.org) smtp.mail=alex+caf_=alex.smola=gmail.com@smola.org;
dkim=pass (test mode) header.i=@googlemail.com
Received: by eaal1 with SMTP id l1so15092746eaa.6
        for <alex.smola@gmail.com>; Tue, 03 Jan 2012 14:17:51 -0800 (PST)
Received: by 10.205.135.18 with SMTP id ie18mr5325064bkc.72.1325629071362;
        Tue, 03 Jan 2012 14:17:51 -0800 (PST)
X-Forwarded-To: alex.smola@gmail.com
X-Forwarded-For: alex@smola.org alex.smola@gmail.com
Delivered-To: alex@smola.org
Received: by 10.204.65.198 with SMTP id k6cs206093bki;
        Tue, 3 Jan 2012 14:17:50 -0800 (PST)
Received: by 10.52.88.179 with SMTP id bh19mr10729402vdb.38.1325629068795;
        Tue, 03 Jan 2012 14:17:48 -0800 (PST)
Return-Path: <althoff.tim@googlemail.com>
Received: from mail-vx0-f179.google.com (mail-vx0-f179.google.com [209.85.220.179])
        by mx.google.com with ESMTPS id dt4si11767074vdb.93.2012.01.03.14.17.48
        (version=TLSv1/SSLv3 cipher=OTHER);
        Tue, 03 Jan 2012 14:17:48 -0800 (PST)
Received-SPF: pass (google.com: domain of althoff.tim@googlemail.com designates
209.85.220.179 as permitted sender) client-ip=209.85.220.179;
Received: by vcbf13 with SMTP id f13so11295098vcb.10
        for <alex@smola.org>; Tue, 03 Jan 2012 14:17:48 -0800 (PST)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
        d=googlemail.com; s=gamma;
        h=mime-version:sender:date:x-google-sender-auth:message-id:subject
         :from:to:content-type;
        bh=WCbdZ5sXac25dpH02XcRyDOdts993hKwsAVXpGrFh0w=;
        b=WK2B2+ExWnf/gvTkw6uUvKuP4XeoKnlJq3USYTm0RARK8dSFjyOQsIHeAP9Yssxp6O
         7ngGoTzYqd+ZsyJfvQcLAWp1PCJhG8AMcnqWkx0NMeoFvIp2HQooZwxSOCx5ZRgY+7qX
         uIbbdna4lUDXj6UFe16SpLDCkptd8OZ3gr7+o=
MIME-Version: 1.0
Received: by 10.220.108.81 with SMTP id e17mr24104004vcp.67.1325629067787;
 Tue, 03 Jan 2012 14:17:47 -0800 (PST)
Sender: althoff.tim@googlemail.com
Received: by 10.220.17.129 with HTTP; Tue, 3 Jan 2012 14:17:47 -0800 (PST)
Date: Tue, 3 Jan 2012 14:17:47 -0800
X-Google-Sender-Auth: 6bwi6D17HjZIkxOEol38NZzyeHs
Message-ID: <CAFJJHDGPBW+SdZg0MdAABiAKydDk9tpeMoDijYGjoGO-WC7osg@mail.gmail.com>
Subject: CS 281B. Advanced Topics in Learning and Decision Making
From: Tim Althoff <althoff@eecs.berkeley.edu>
To: alex@smola.org
Content-Type: multipart/alternative; boundary=f46d043c7af4b07e8d04b5a7113a

--f46d043c7af4b07e8d04b5a7113a
Content-Type: text/plain; charset=ISO-8859-1
```

Feature Engineering for Spam Filtering

- bag of words
- pairs of words
- date & time
- recipient path
- IP number
- sender
- encoding
- links
- ... secret sauce ...

Carnegie Mellon University

More feature engineering

- Two Interlocking Spirals
  Transform the data into a radial and angular part
  $$(x_1, x_2) = (r \sin \phi, r \cos \phi)$$
- Handwritten Japanese Character Recognition
  - Break down the images into strokes and recognize it
  - Lookup based on stroke order
- Medical Diagnosis
  - Physician's comments
  - Blood status / ECG / height / weight / temperature ...
  - Medical knowledge
- Preprocessing
  - Zero mean, unit variance to fix scale issue (e.g. weight vs. income)
  - Probability integral transform (inverse CDF) as alternative

The Perceptron on features

initialize $w, b = 0$

repeat

Pick (x_i, y_i) from data

if $y_i(w \cdot \Phi(x_i) + b) \leq 0$ then

$$w' = w + y_i \Phi(x_i)$$

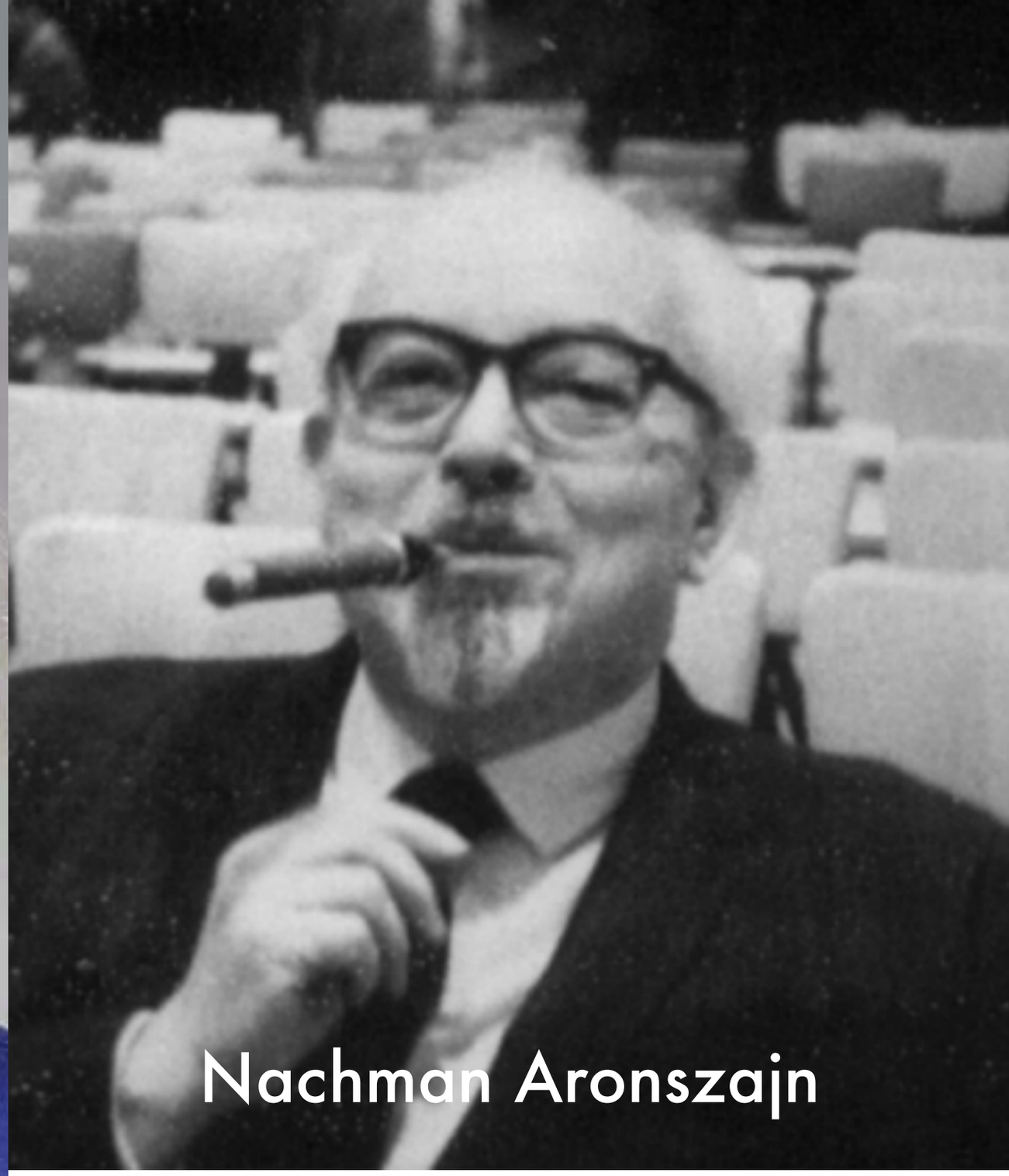
$$b' = b + y_i$$

until $y_i(w \cdot \Phi(x_i) + b) > 0$ for all i

- Nothing happens if classified correctly
- Weight vector is linear combination $w = \sum_{i \in I} y_i \phi(x_i)$
- Classifier is linear combination of inner products $f(x) = \sum_{i \in I} y_i \langle \phi(x_i), \phi(x) \rangle + b$

Problems

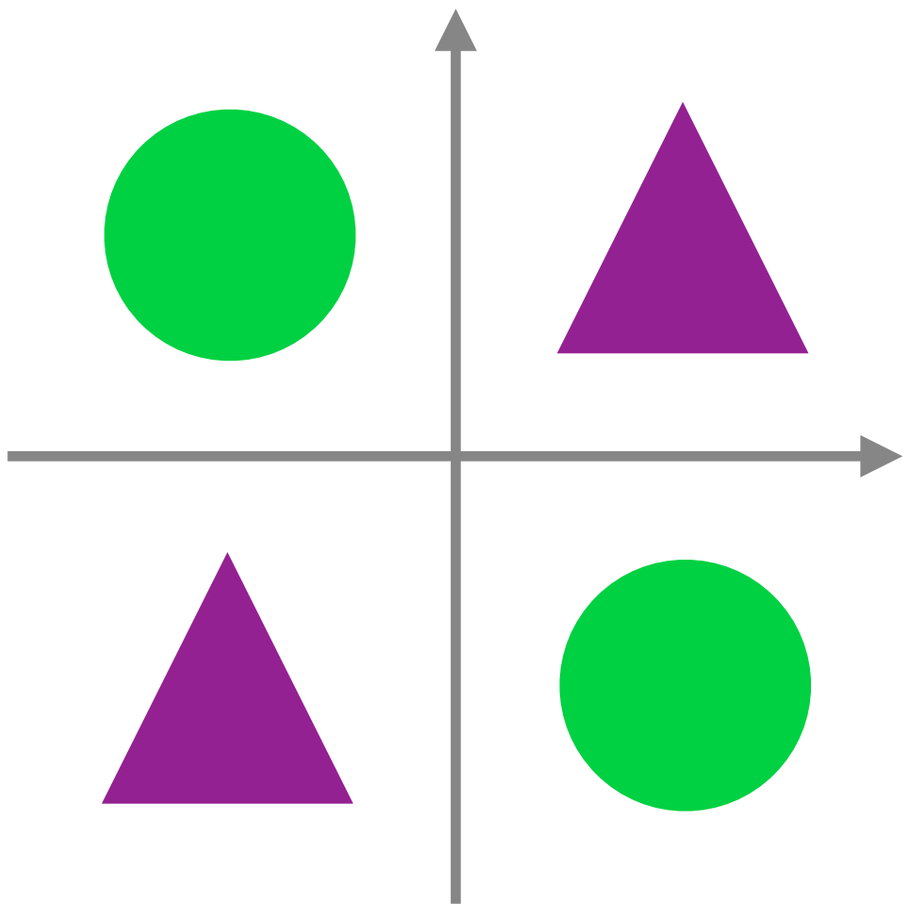
- Problems
  - Need domain expert (e.g. Chinese OCR)
  - Often expensive to compute
  - Difficult to transfer engineering knowledge
- Shotgun Solution
  - Compute many features
  - Hope that this contains good ones
  - Do this efficiently
- Nonlinear methods (needs lots of data & cpu) learn the features and the classifier



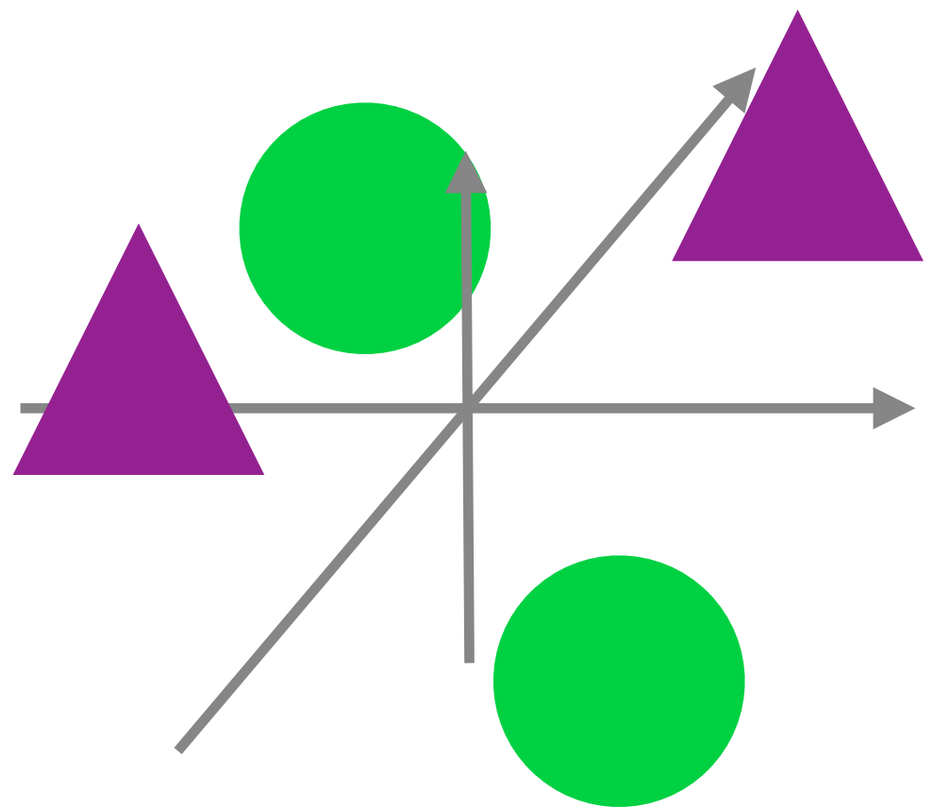
Kernels

Carnegie Mellon University

Solving XOR



(x_1, x_2)



$(x_1, x_2, x_1 x_2)$

- XOR not linearly separable
- Mapping into 3 dimensions makes it easily solvable

Quadratic Features

Quadratic Features in $\mathbb{R}^2$

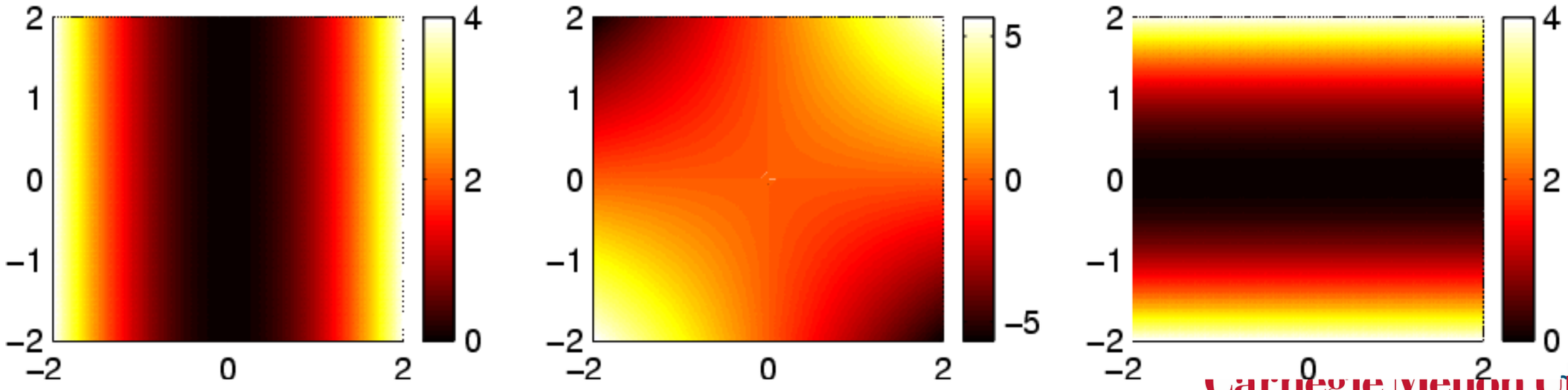
$$\Phi(x) := \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$$

Dot Product

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

Insight

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



SVM with a polynomial Kernel visualization

Created by:
Udi Aharoni

SVM with a polynomial Kernel visualization

Created by:
Udi Aharoni

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 $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ 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 $\Phi(x)$ ...

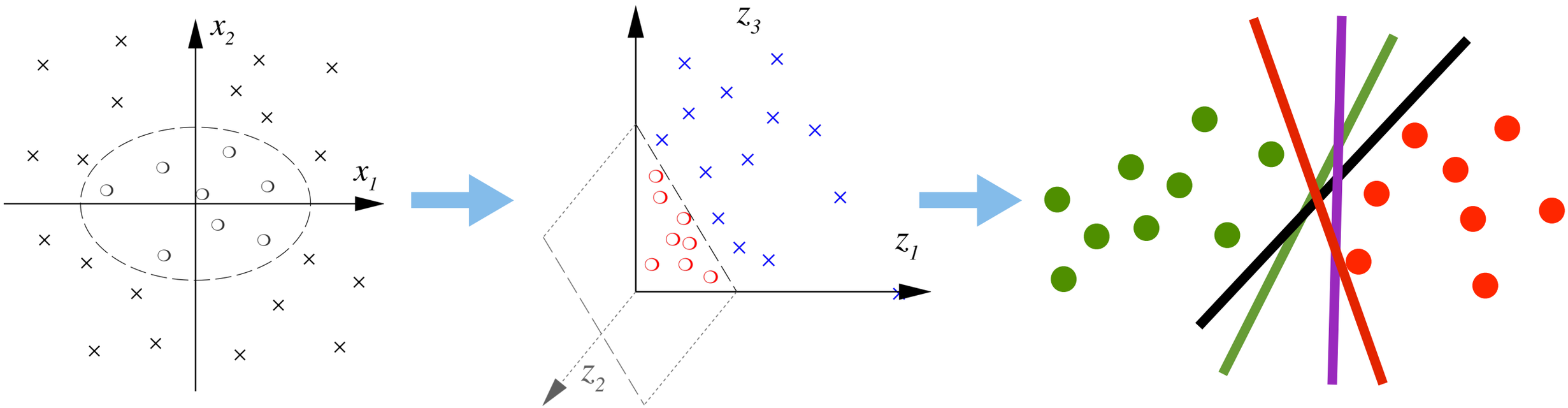
The Kernel Perceptron

initialize $f = 0$
repeat
    Pick (x_i, y_i) from data
    if $y_i f(x_i) \leq 0$ then
        $f(\cdot) \leftarrow f(\cdot) + y_i k(x_i, \cdot) + y_i$
until $y_i f(x_i) > 0$ for all i

- Nothing happens if classified correctly
- Weight vector is linear combination $w = \sum_{i \in I} y_i \phi(x_i)$
- Classifier is linear combination of inner products

$$f(x) = \sum_{i \in I} y_i \langle \phi(x_i), \phi(x) \rangle + b = \sum_{i \in I} y_i k(x_i, x) + b$$

Processing Pipeline



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

Polynomial Kernels

Idea

- We want to extend $k(x, x') = \langle x, x' \rangle^2$ to

$$k(x, x') = (\langle x, x' \rangle + c)^d \text{ where } c > 0 \text{ and } d \in \mathbb{N}.$$

- Prove that such a kernel corresponds to a dot product.

Proof strategy

Simple and straightforward: compute the explicit sum given by the kernel, i.e.

$$k(x, x') = (\langle x, x' \rangle + c)^d = \sum_{i=0}^{m} \binom{d}{i} (\langle x, x' \rangle)^i \, c^{d-i}$$

Individual terms $(\langle x, x' \rangle)^i$ are dot products for some $\Phi_i(x)$.

Kernel Conditions

Computability

We have to be able to compute $k(x, x')$ efficiently (much cheaper than dot products themselves).

"Nice and Useful" Functions

The features themselves have to be useful for the learning problem at hand. Quite often this means smooth functions.

Symmetry

Obviously $k(x, x') = k(x', x)$ due to the symmetry of the dot product $\langle \Phi(x), \Phi(x') \rangle = \langle \Phi(x'), \Phi(x) \rangle$.

Dot Product in Feature Space

Is there always a Φ such that k really is a dot product?

Mercer's Theorem

The Theorem

For any symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which is square integrable in $\mathcal{X} \times \mathcal{X}$ and which satisfies

$$\int_{\mathcal{X}\times\mathcal{X}} k(x, x')f(x)f(x')dxdx' \geq 0 \text{ for all } f \in L_2(\mathcal{X})$$

there exist $\phi_i : \mathcal{X} \to \mathbb{R}$ and numbers $\lambda_i \geq 0$ where

$$k(x, x') = \sum_i \lambda_i \phi_i(x)\phi_i(x') \text{ for all } x, x' \in \mathcal{X}.$$

Interpretation

Double integral is the continuous version of a vector-matrix-vector multiplication. For positive semidefinite matrices we have

$$\sum\sum k(x_i, x_j)\alpha_i\alpha_j \geq 0$$

Properties

Distance in Feature Space

Distance between points in feature space via

$$\begin{aligned} d(x,x')^2 :=& \|\Phi(x) - \Phi(x')\|^2 \\ =& \langle \Phi(x), \Phi(x) \rangle - 2\langle \Phi(x), \Phi(x') \rangle + \langle \Phi(x'), \Phi(x') \rangle \\ =& k(x,x) + k(x',x') - 2k(x,x) \end{aligned}$$

Kernel Matrix

To compare observations we compute dot products, so we study the matrix K given by

$$K_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle = k(x_i, x_j)$$

where x_i are the training patterns.

Similarity Measure

The entries K_{ij} tell us the overlap between $\Phi(x_i)$ and $\Phi(x_j)$, so $k(x_i, x_j)$ is a similarity measure.

Properties

K is Positive Semidefinite

Claim: $\alpha^\top K \alpha \geq 0$ for all $\alpha \in \mathbb{R}^m$ and all kernel matrices $K \in \mathbb{R}^{m \times m}$. Proof:

$$\sum_{i,j}^{m} \alpha_i \alpha_j K_{ij} = \sum_{i,j}^{m} \alpha_i \alpha_j \langle \Phi(x_i), \Phi(x_j) \rangle$$

$$= \left\langle \sum_i^m \alpha_i \Phi(x_i), \sum_j^m \alpha_j \Phi(x_j) \right\rangle = \left\| \sum_{i=1}^m \alpha_i \Phi(x_i) \right\|^2$$

Kernel Expansion

If w is given by a linear combination of $\Phi(x_i)$ we get

$$\langle w, \Phi(x) \rangle = \left\langle \sum_{i=1}^m \alpha_i \Phi(x_i), \Phi(x) \right\rangle = \sum_{i=1}^m \alpha_i k(x_i, x).$$

A Counterexample

A Candidate for a Kernel

$$k(x, x') = \begin{cases} 1 & \text{if } \|x - x'\| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

This is symmetric and gives us some information about the proximity of points, yet it is not a proper kernel . . .

Kernel Matrix

We use three points, $x_1 = 1, x_2 = 2, x_3 = 3$ and compute the resulting "kernelmatrix" K. This yields

$$K = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$ and eigenvalues $(\sqrt{2}-1)^{-1}, 1$ and $(1-\sqrt{2})$.

as eigensystem. Hence k is not a kernel.

Examples

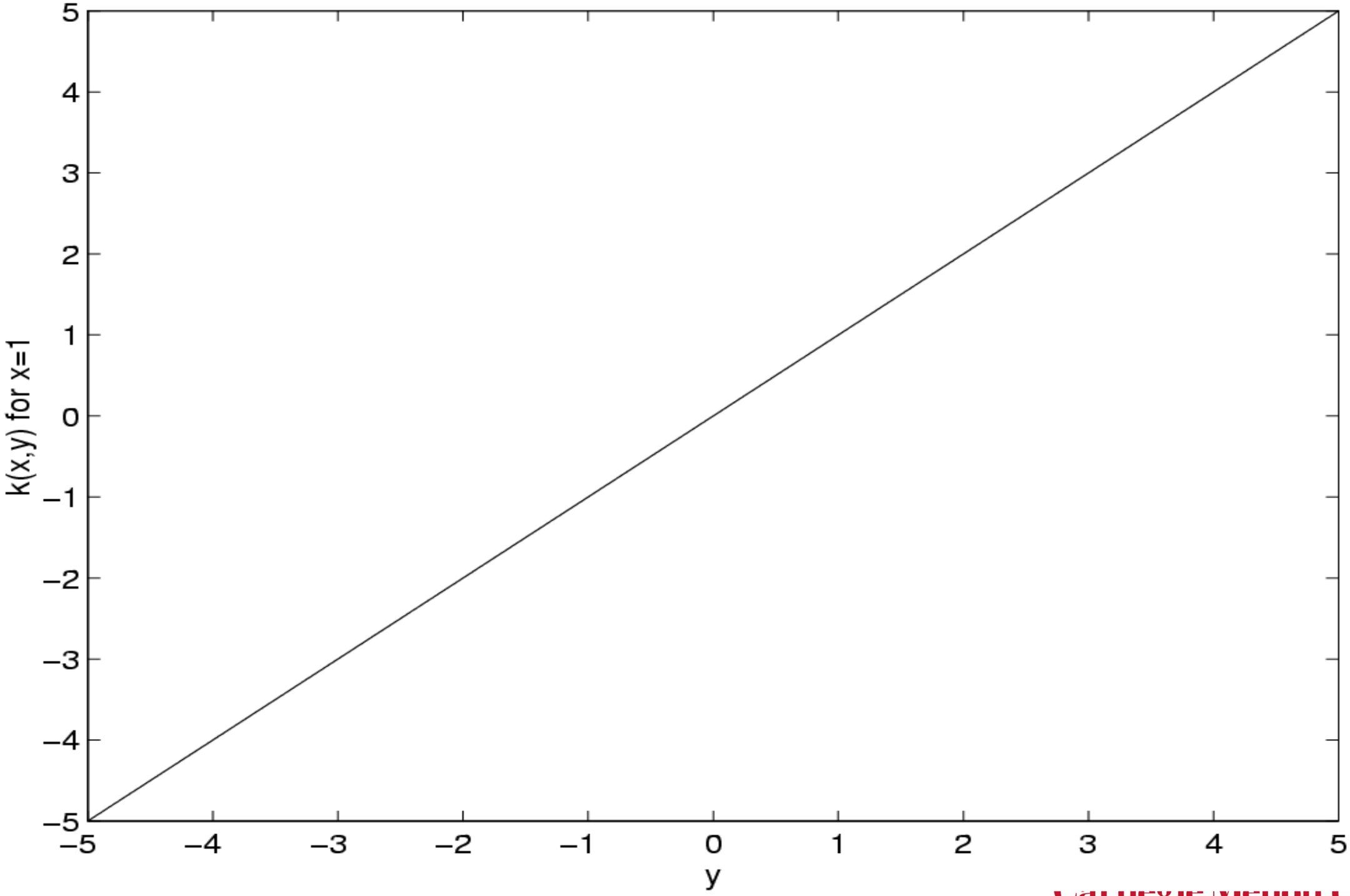
Examples of kernels $k(x, x')$

Linear	$\langle x, x' \rangle$
Laplacian RBF	$\exp\left(-\lambda \|x - x'\|\right)$
Gaussian RBF	$\exp\left(-\lambda \|x - x'\|^2\right)$
Polynomial	$\left(\langle x, x' \rangle + c\rangle\right)^d, c \geq 0,\ d \in \mathbb{N}$
B-Spline	$B_{2n+1}(x - x')$
Cond. Expectation	$\mathbf{E}_c[p(x\|c)p(x'\|c)]$

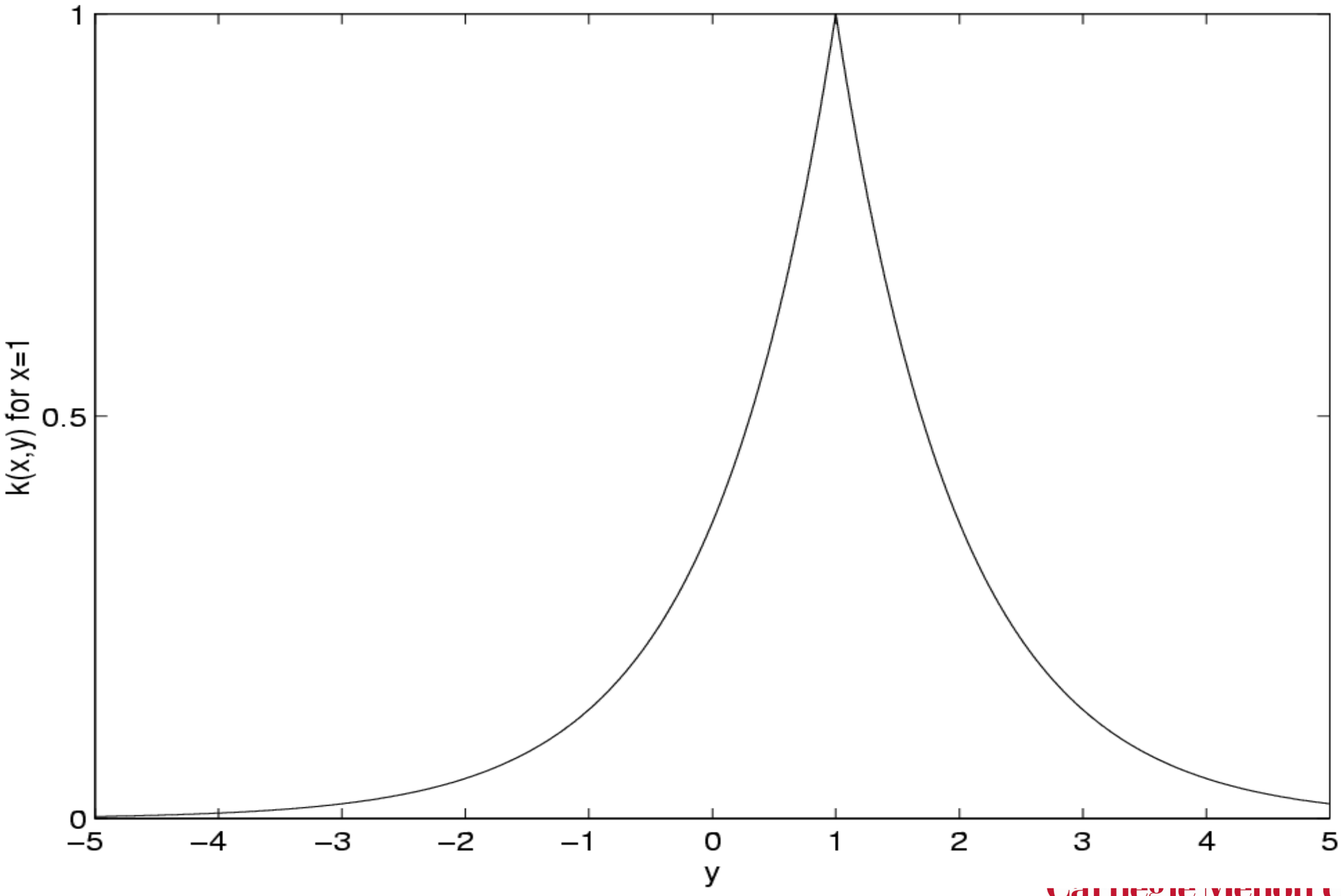
Simple trick for checking Mercer's condition

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

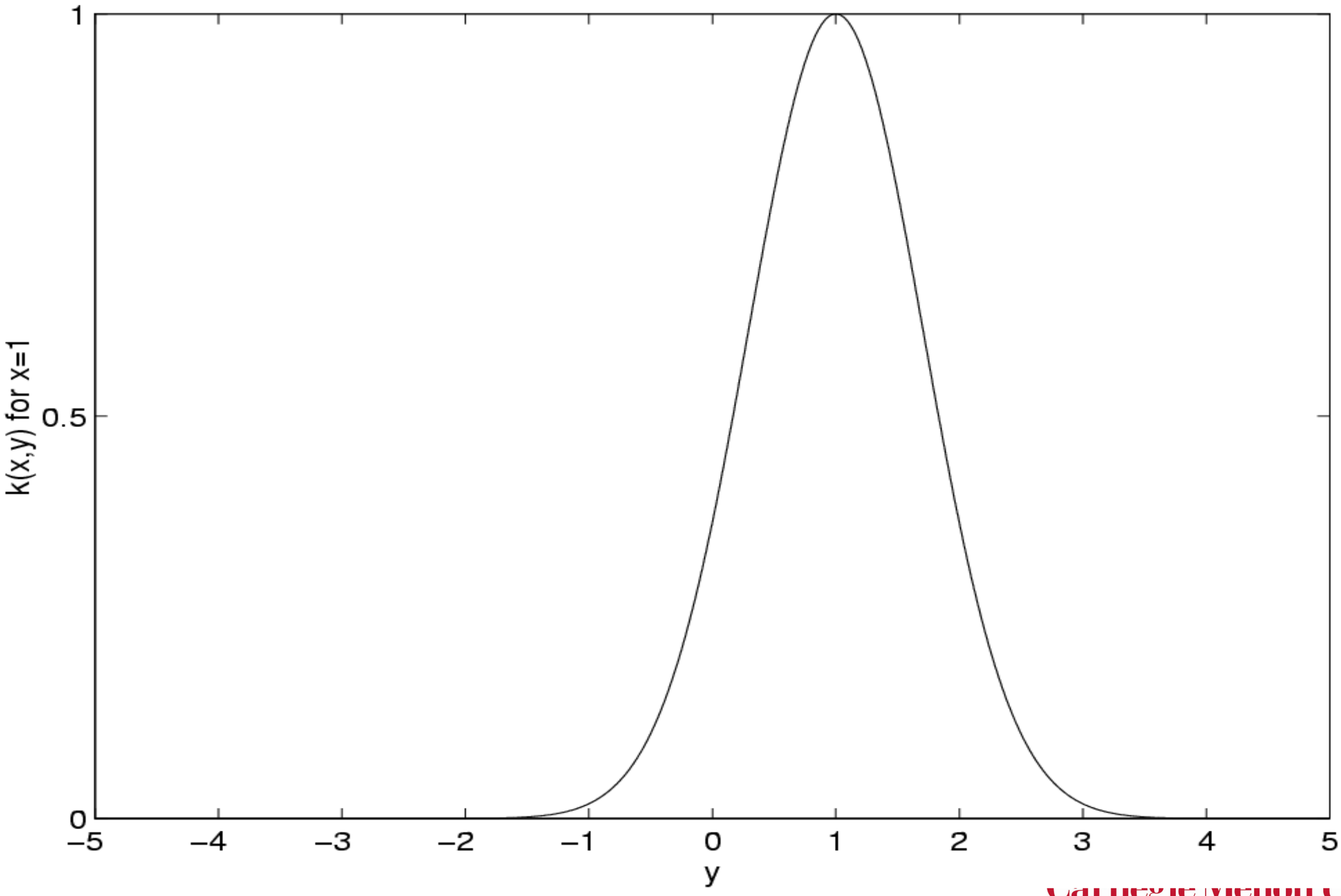
Linear Kernel



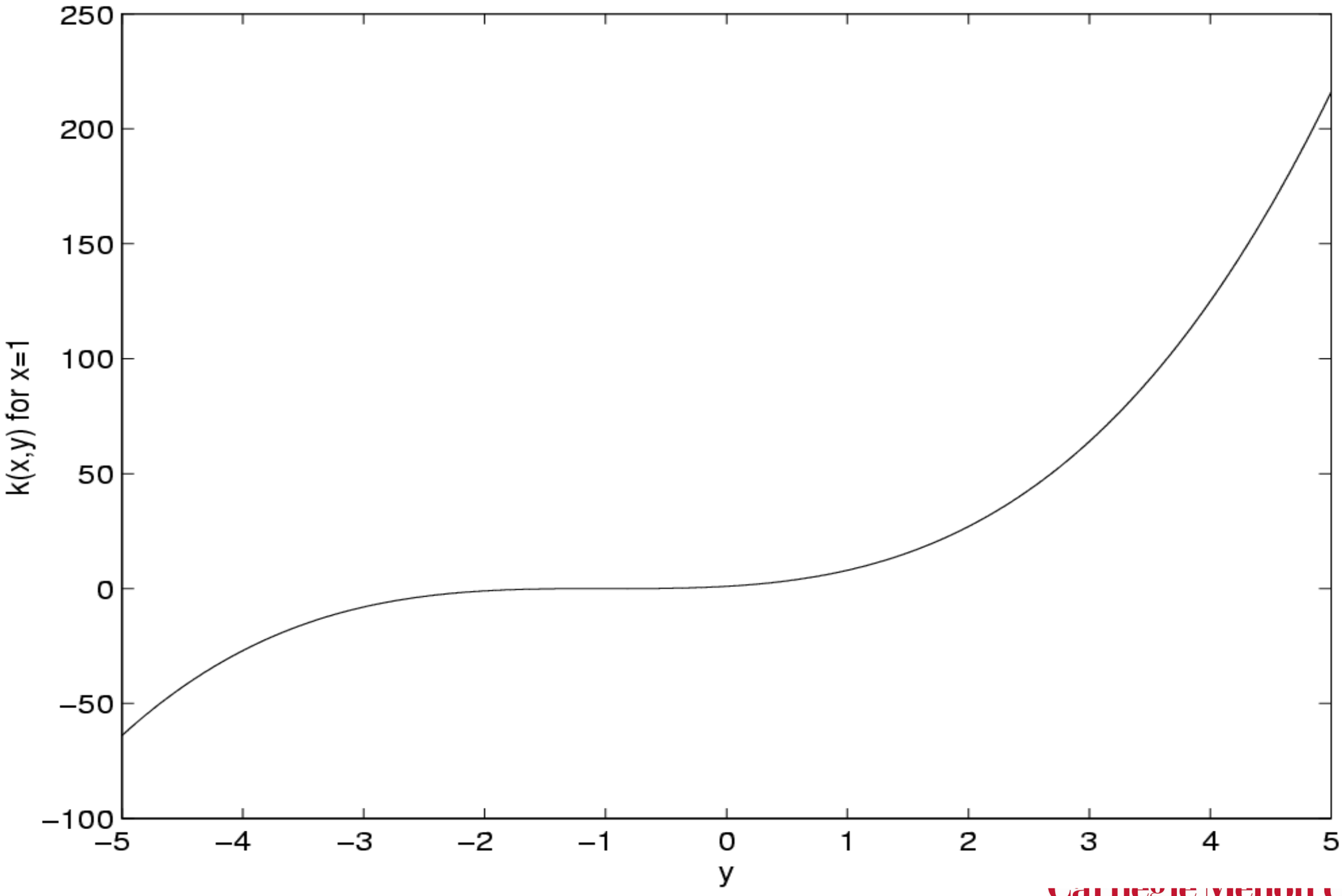
Laplacian Kernel



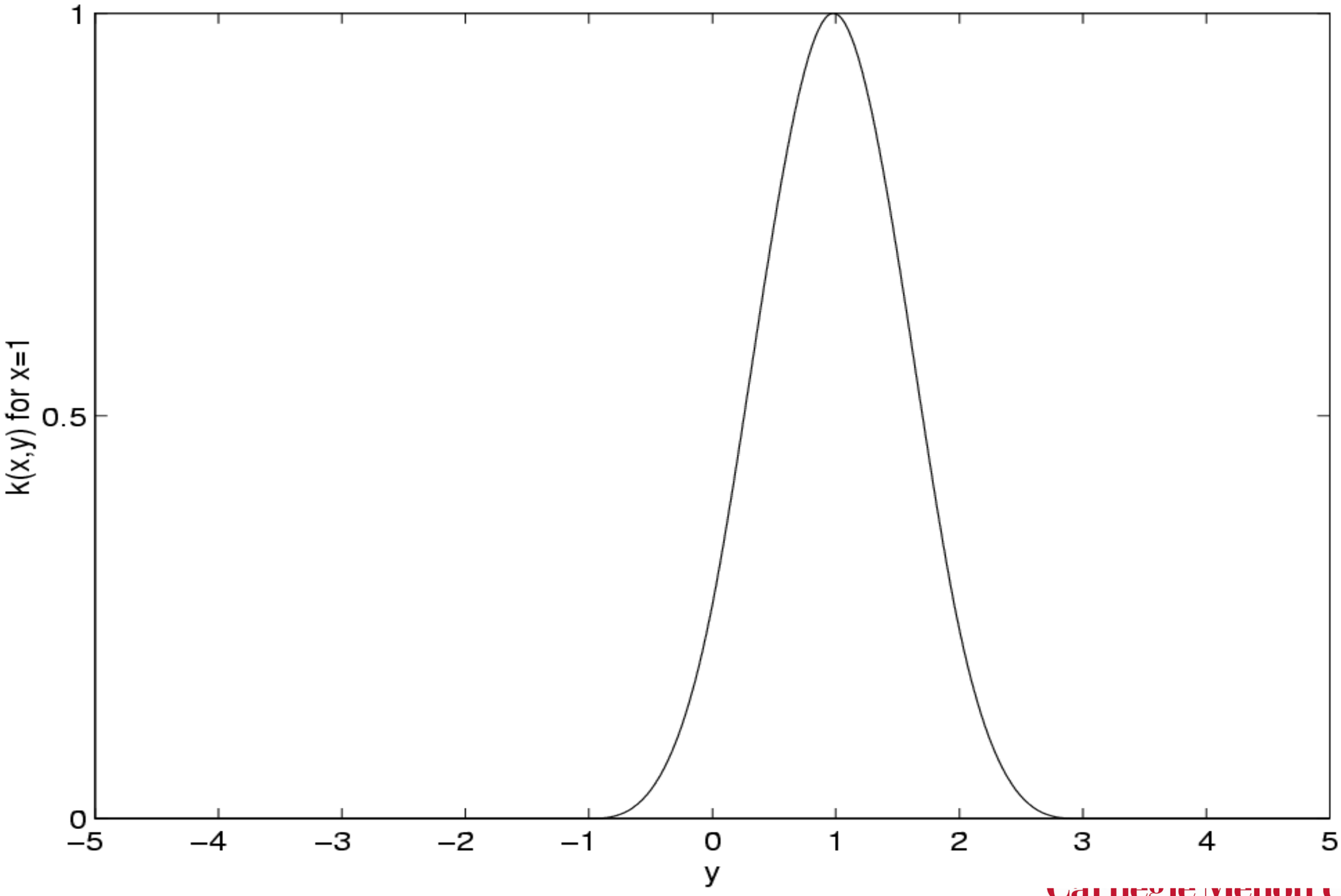
Gaussian Kernel

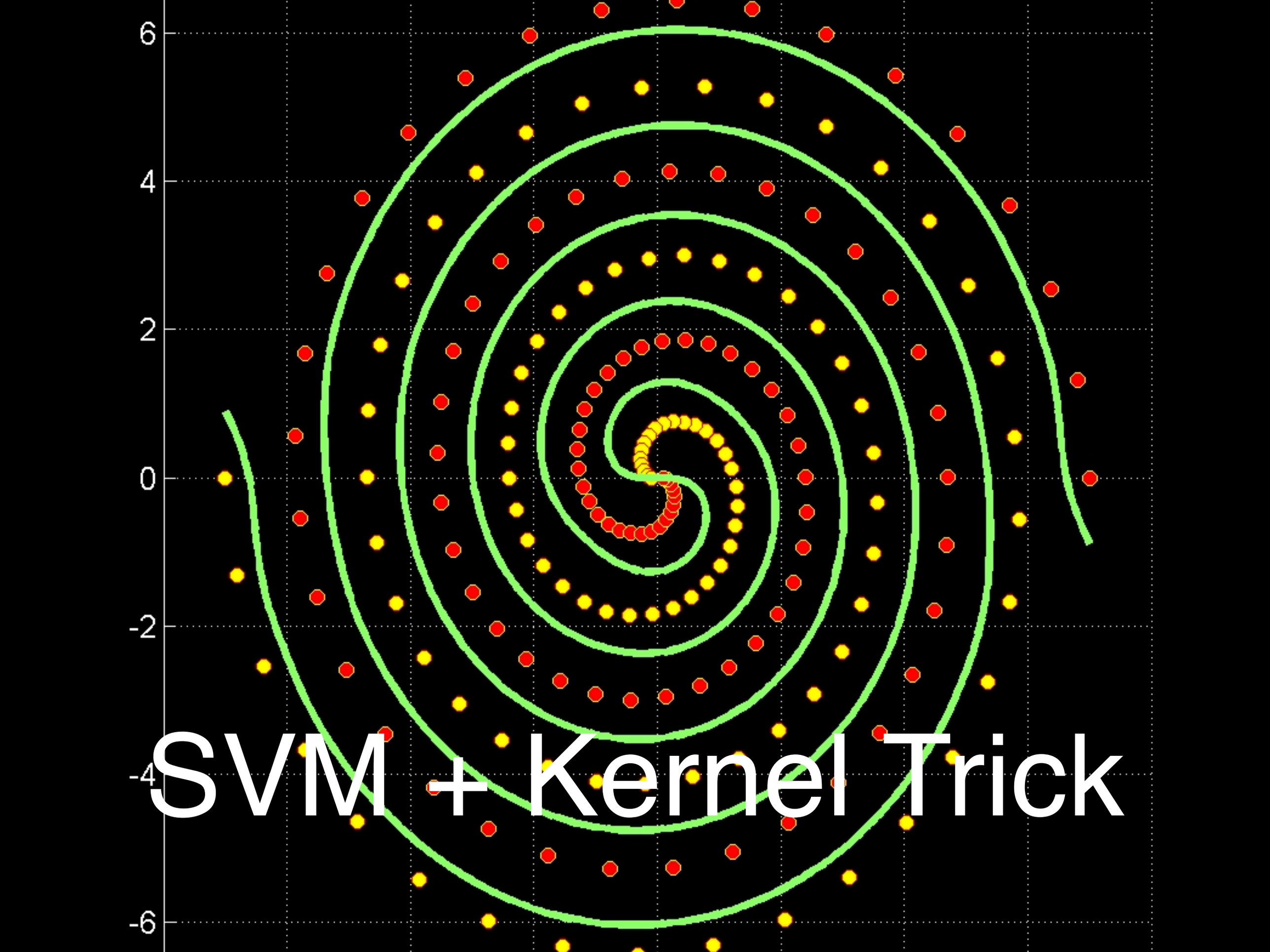


Polynomial of order 3



B_3 Spline Kernel





SVM + Kernel Trick

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

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

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

- Support vector expansion

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

The Kernel Trick

- Linear soft margin problem

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

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

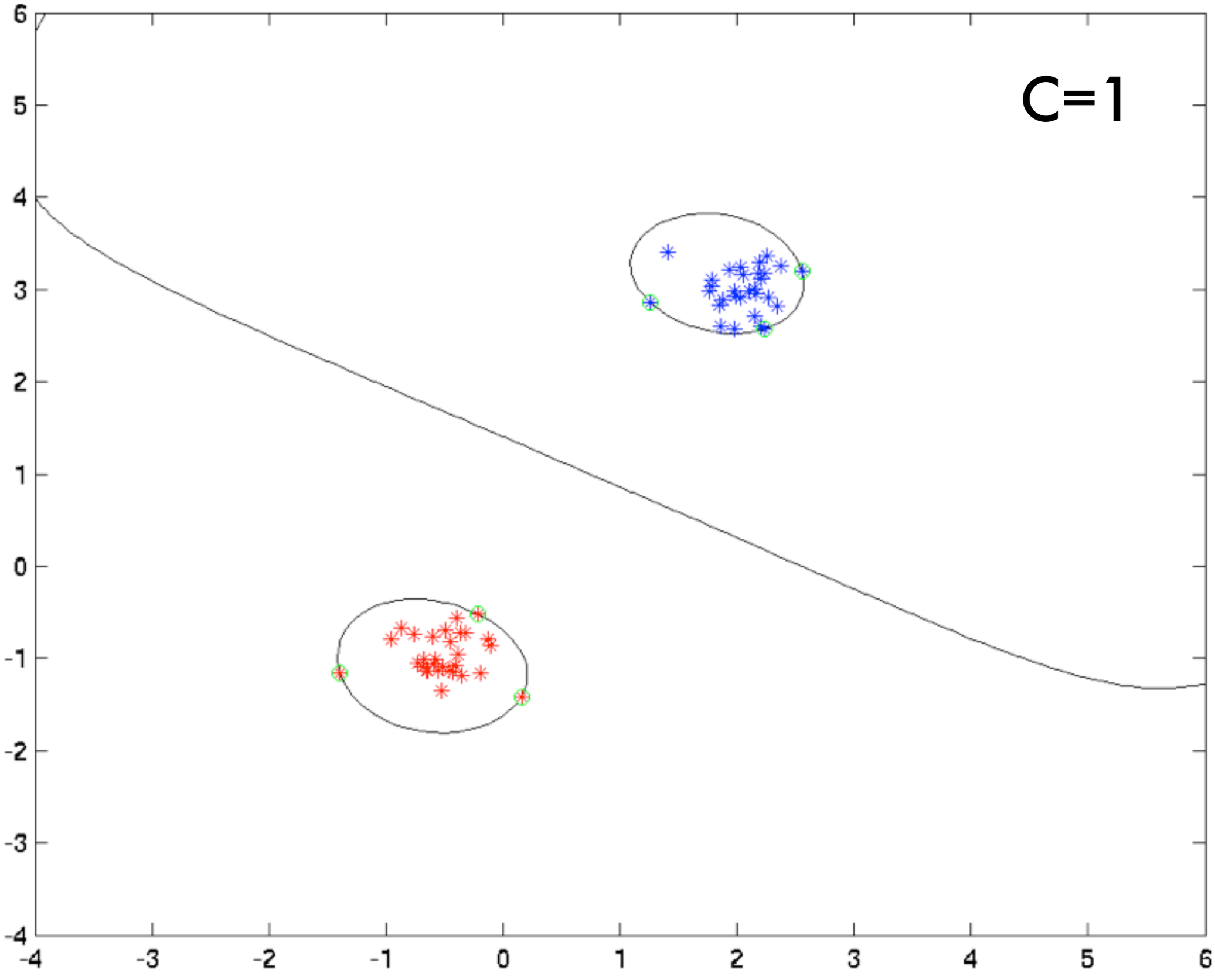
- Dual problem

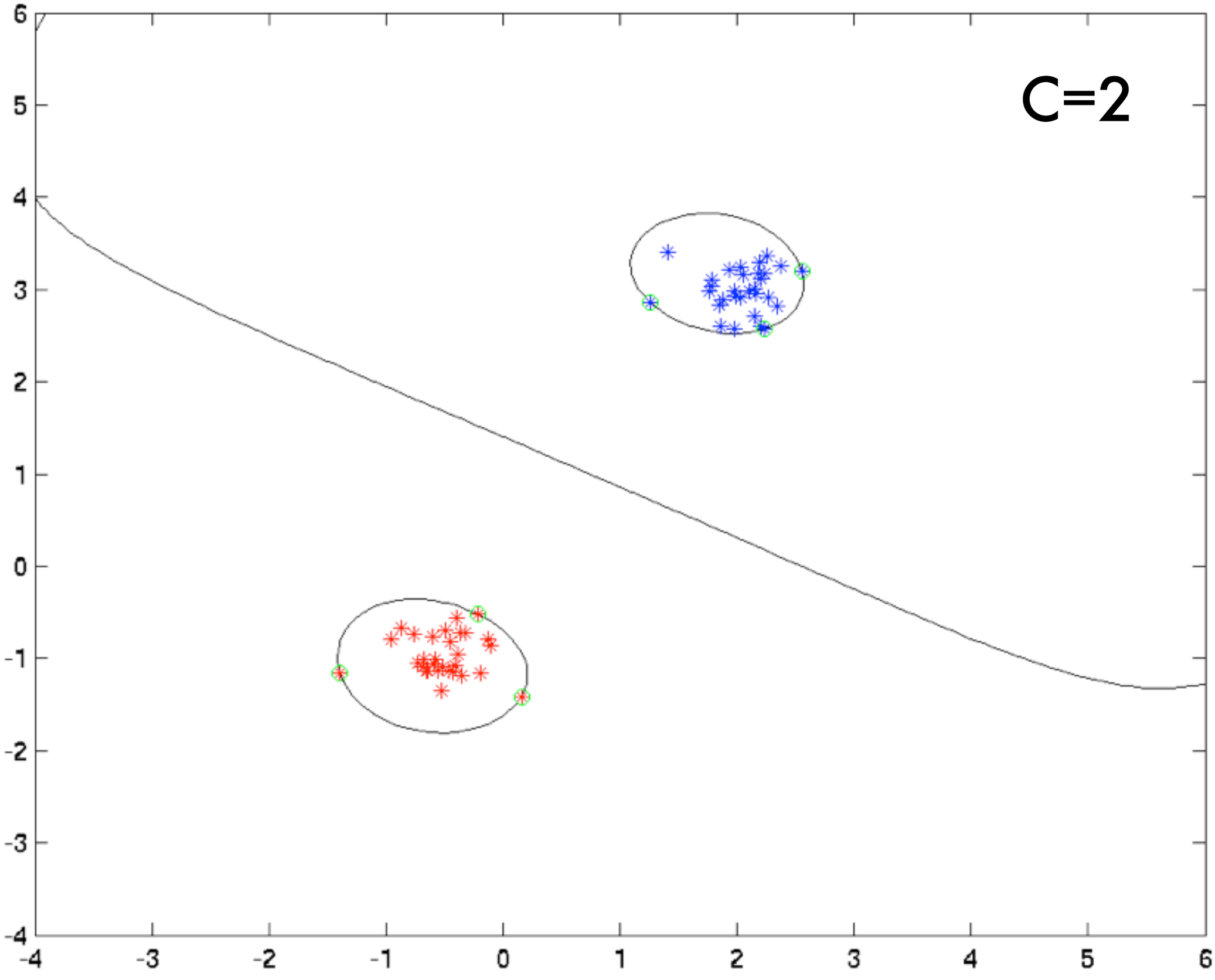
$$\underset{\alpha}{\text{maximize}} -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j k(x_i, x_j) + \sum_i \alpha_i$$

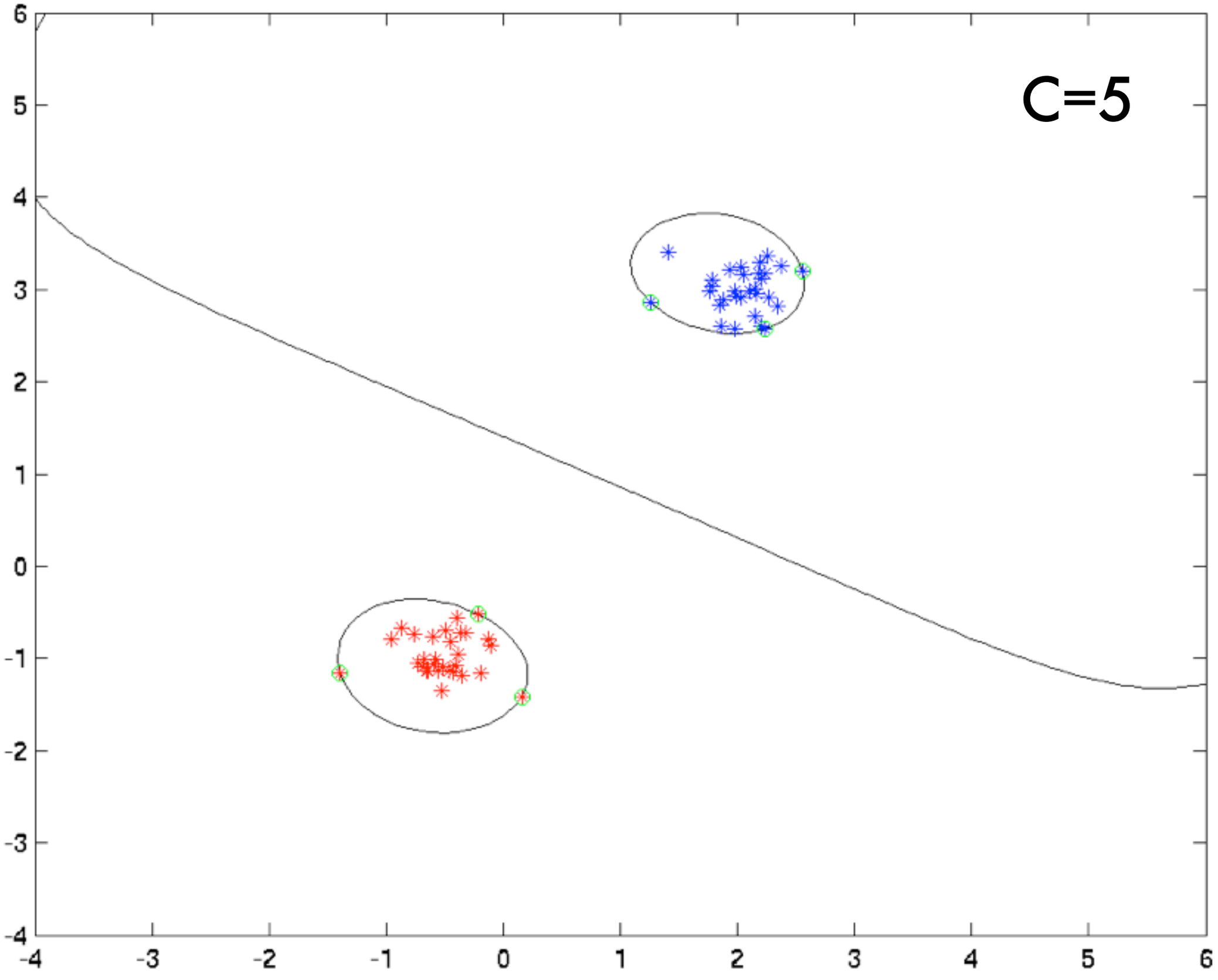
$$\text{subject to } \sum_i \alpha_i y_i = 0 \text{ and } \alpha_i \in [0, C]$$

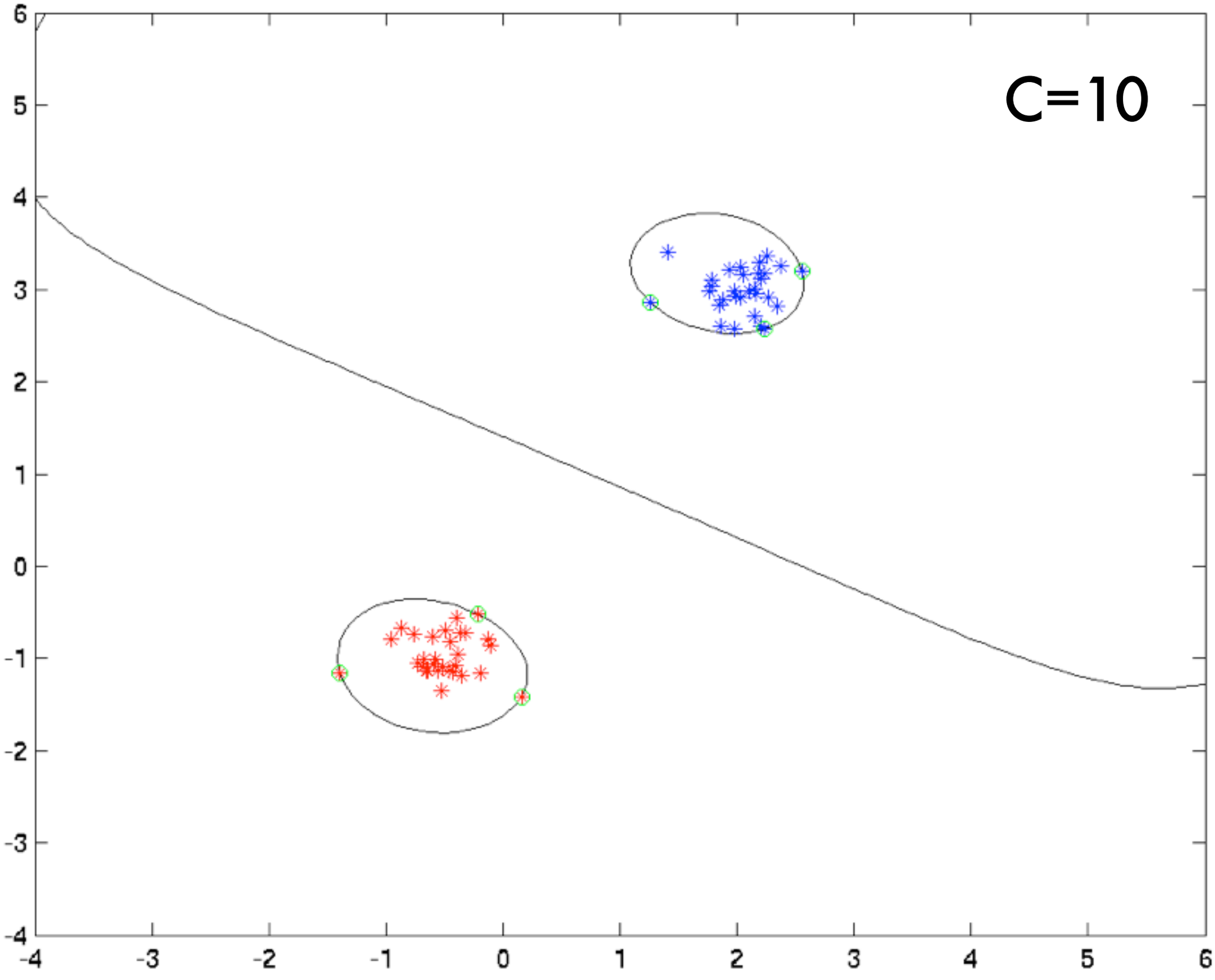
- Support vector expansion

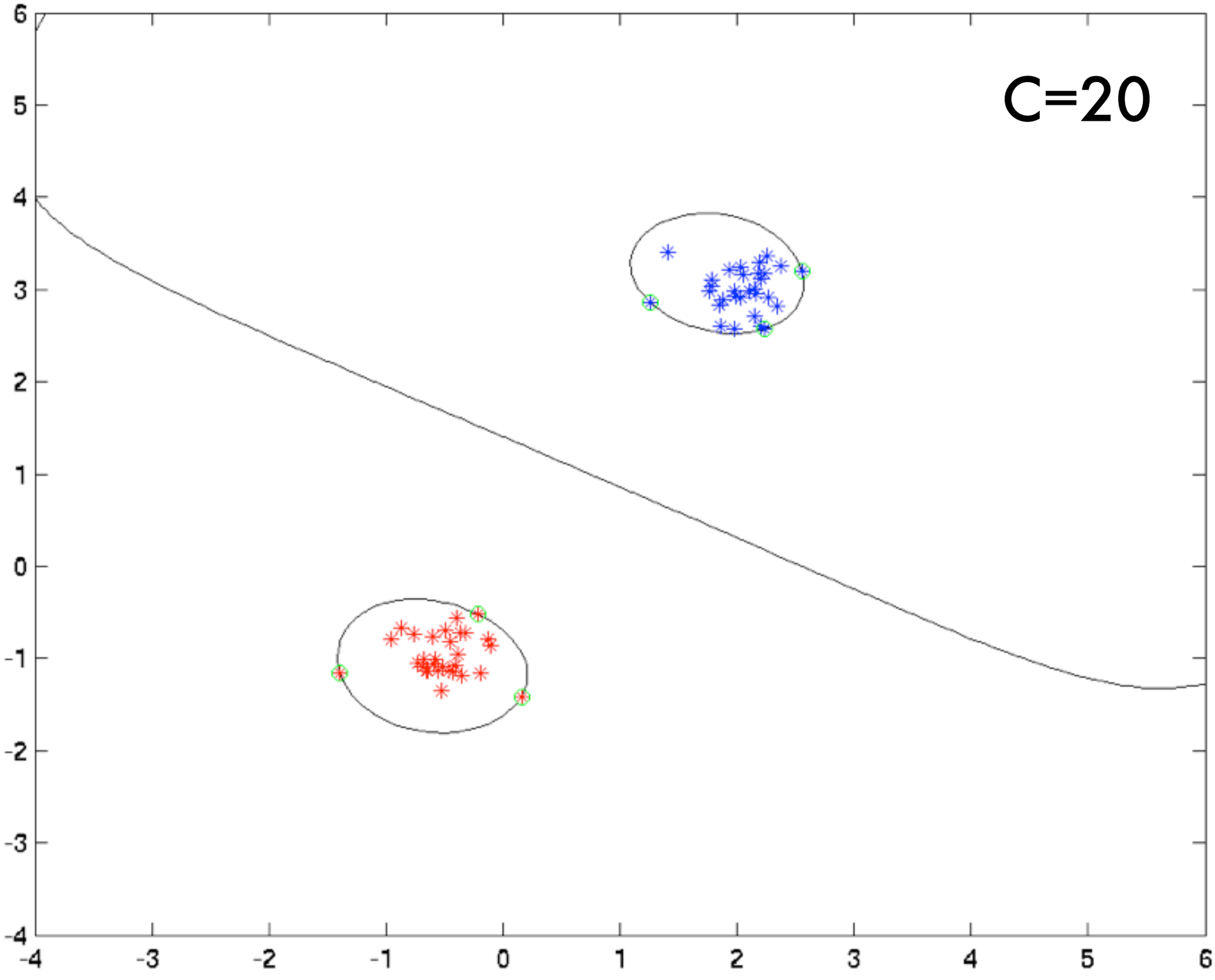
$$f(x) = \sum_i \alpha_i y_i k(x_i, x) + b$$

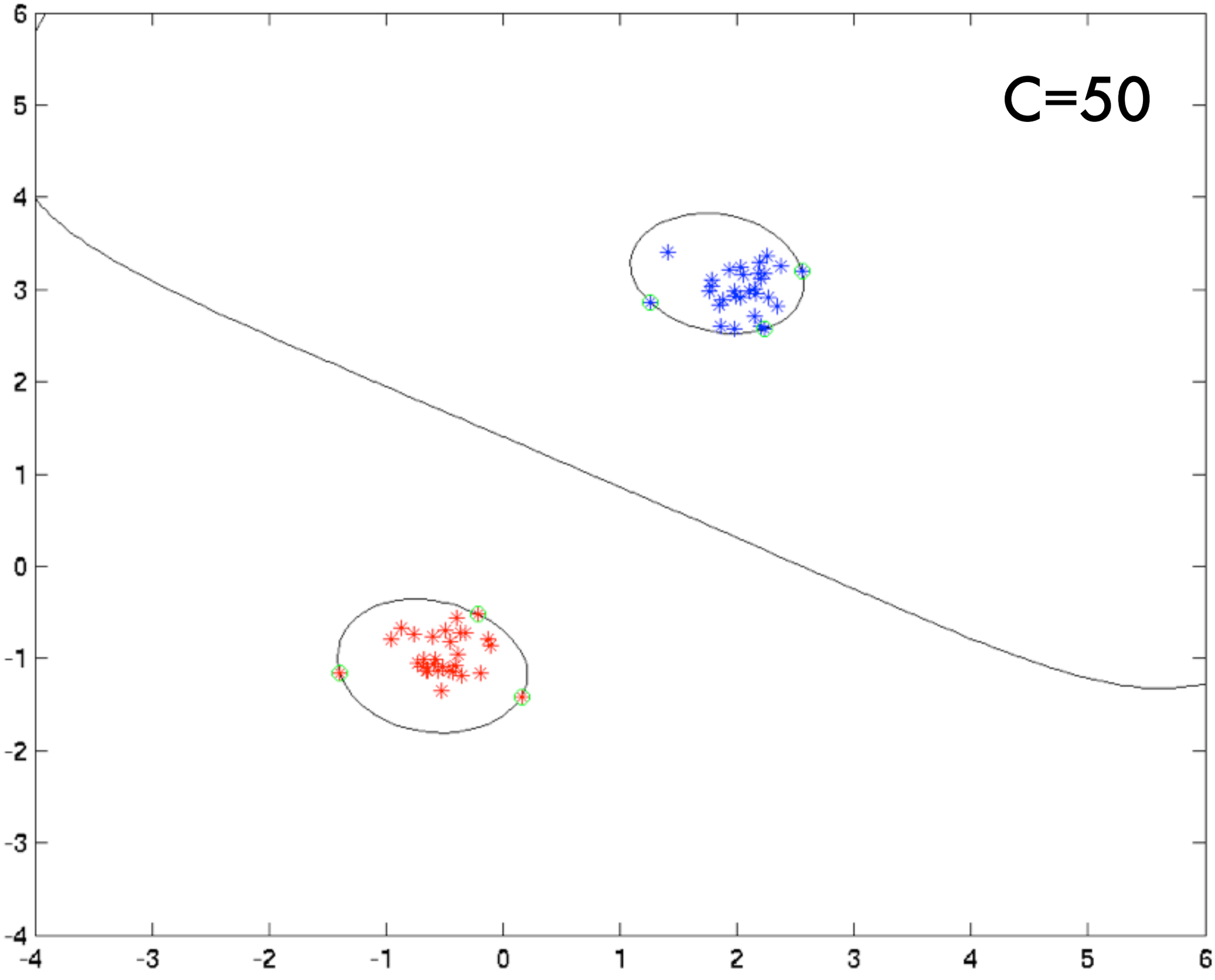

C=1
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

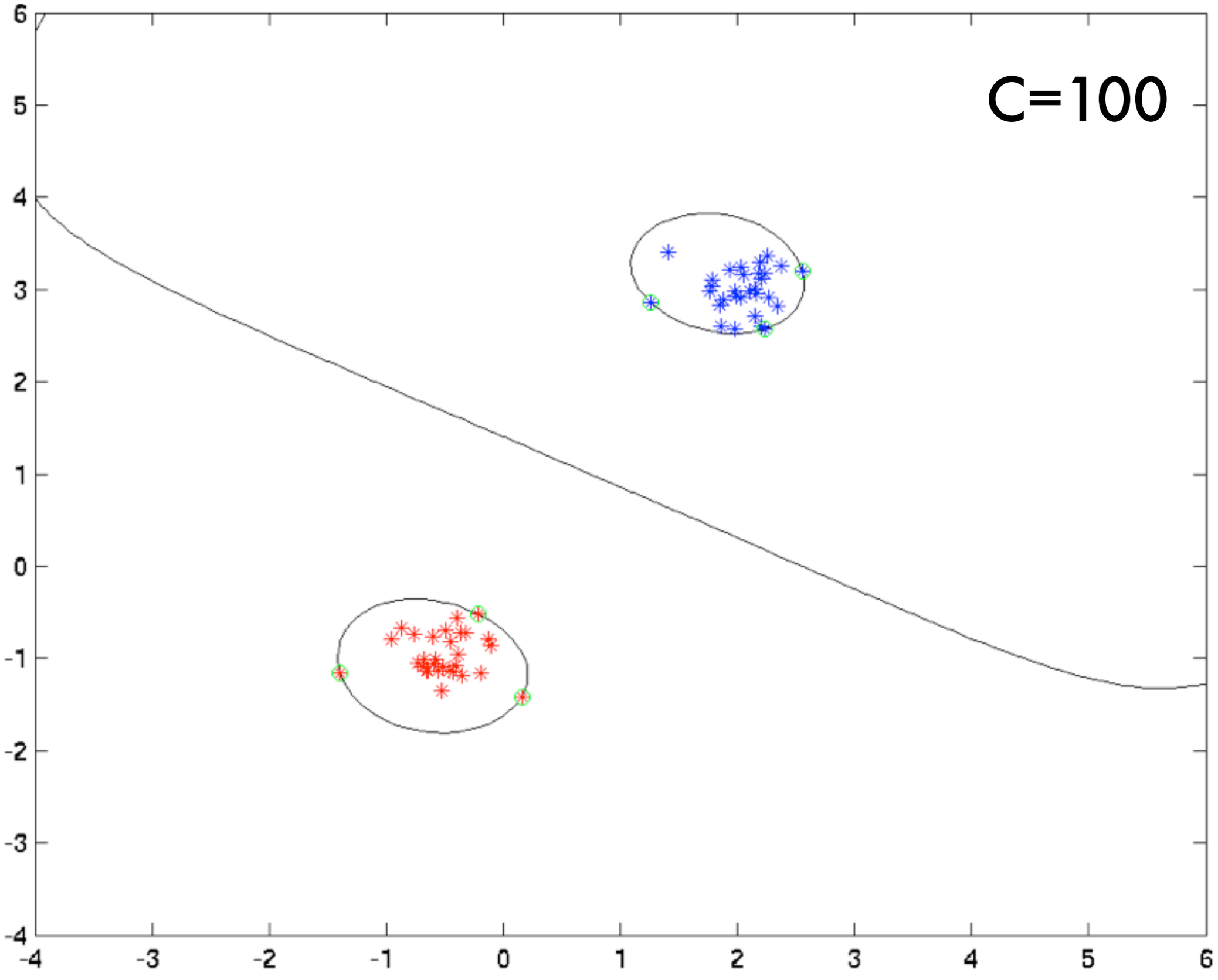

C=2
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

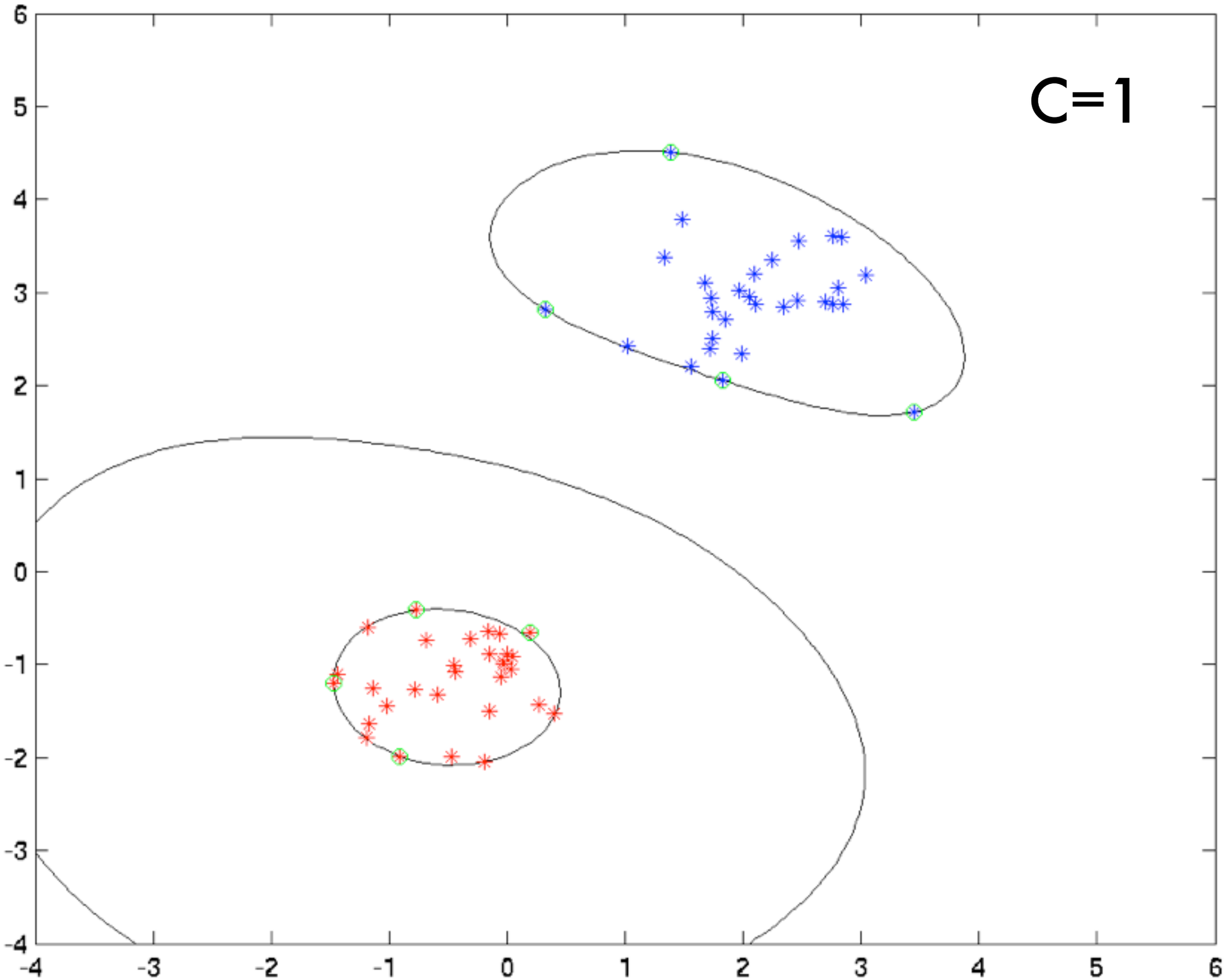

C=5
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

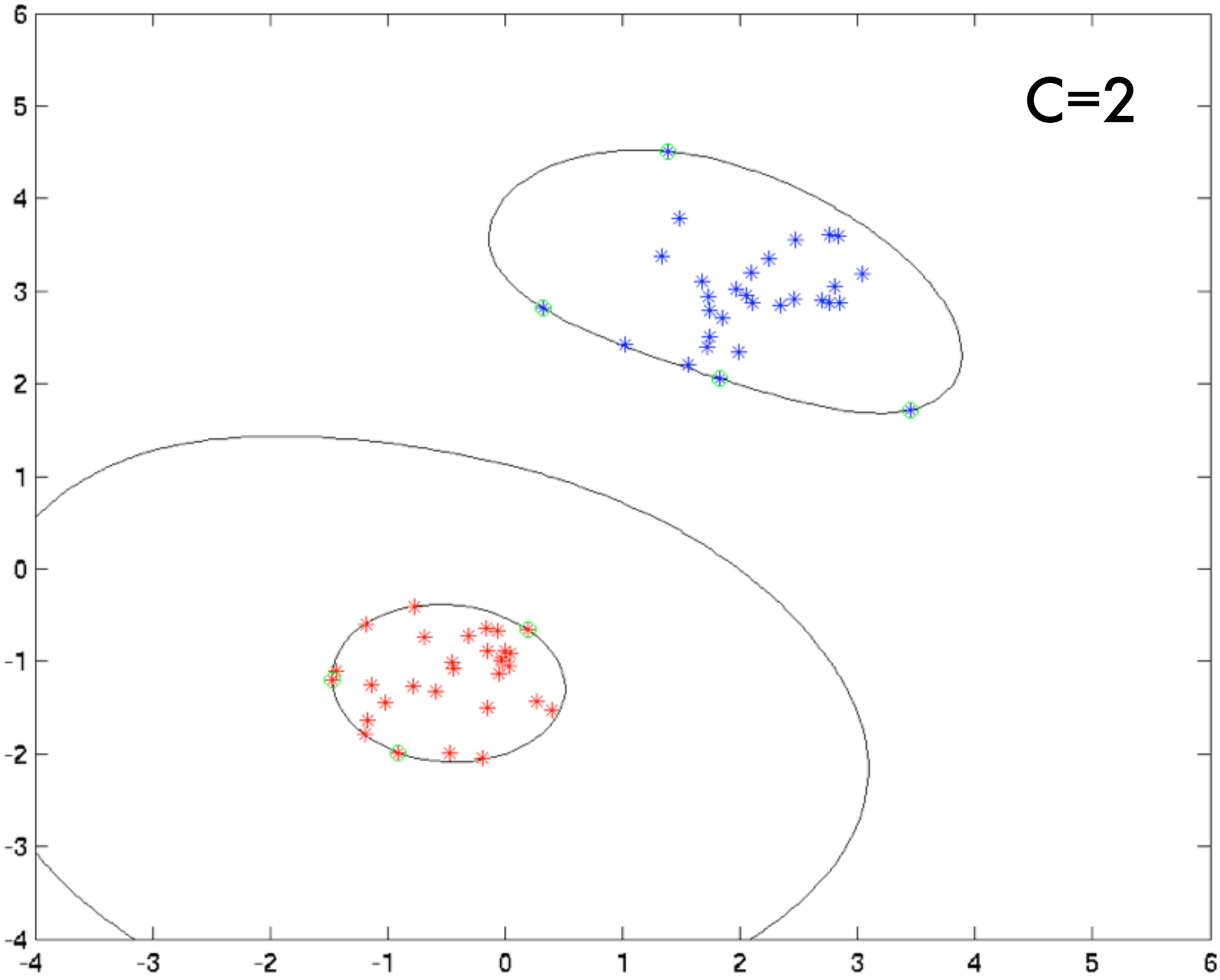

C=10
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

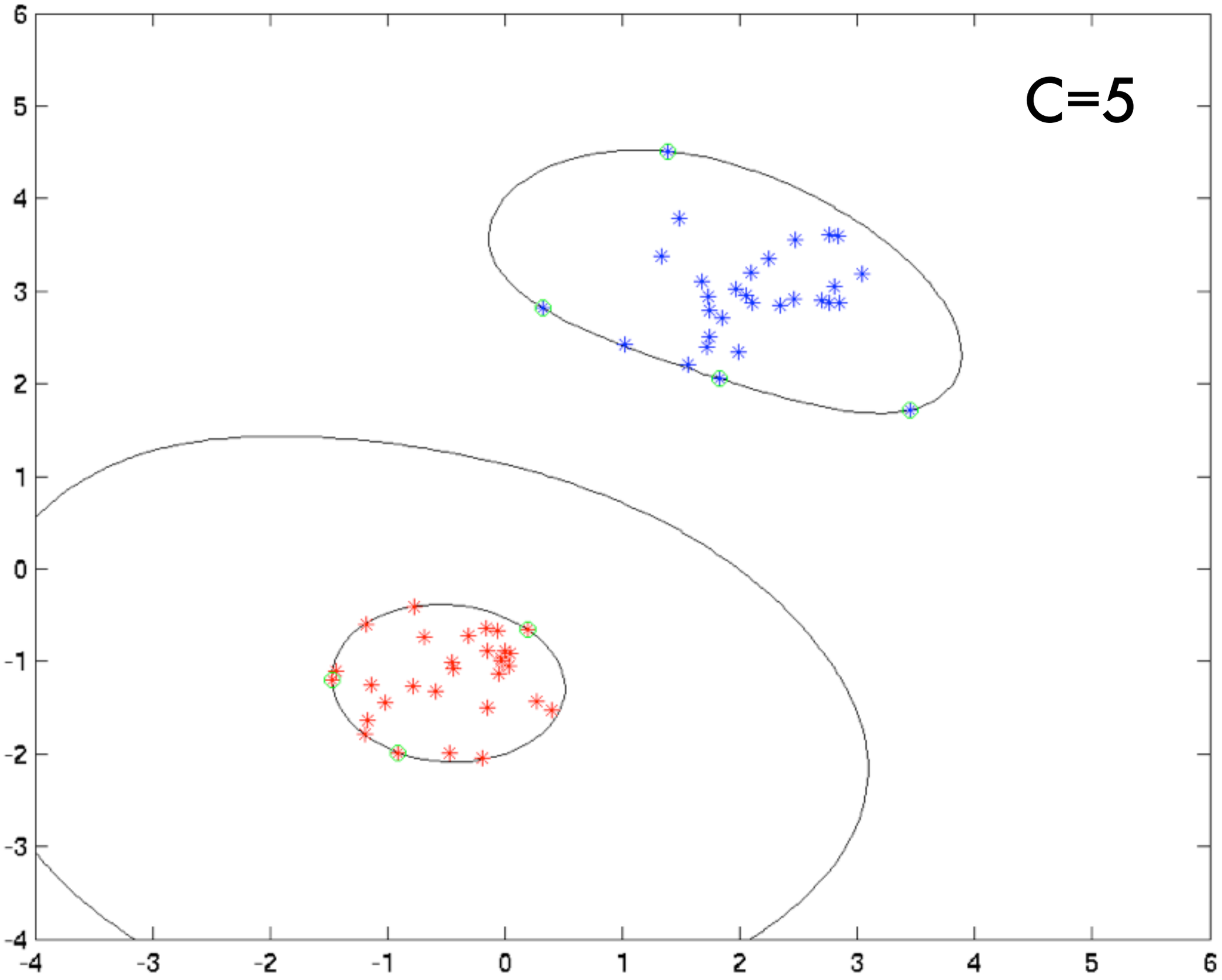

C=20
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

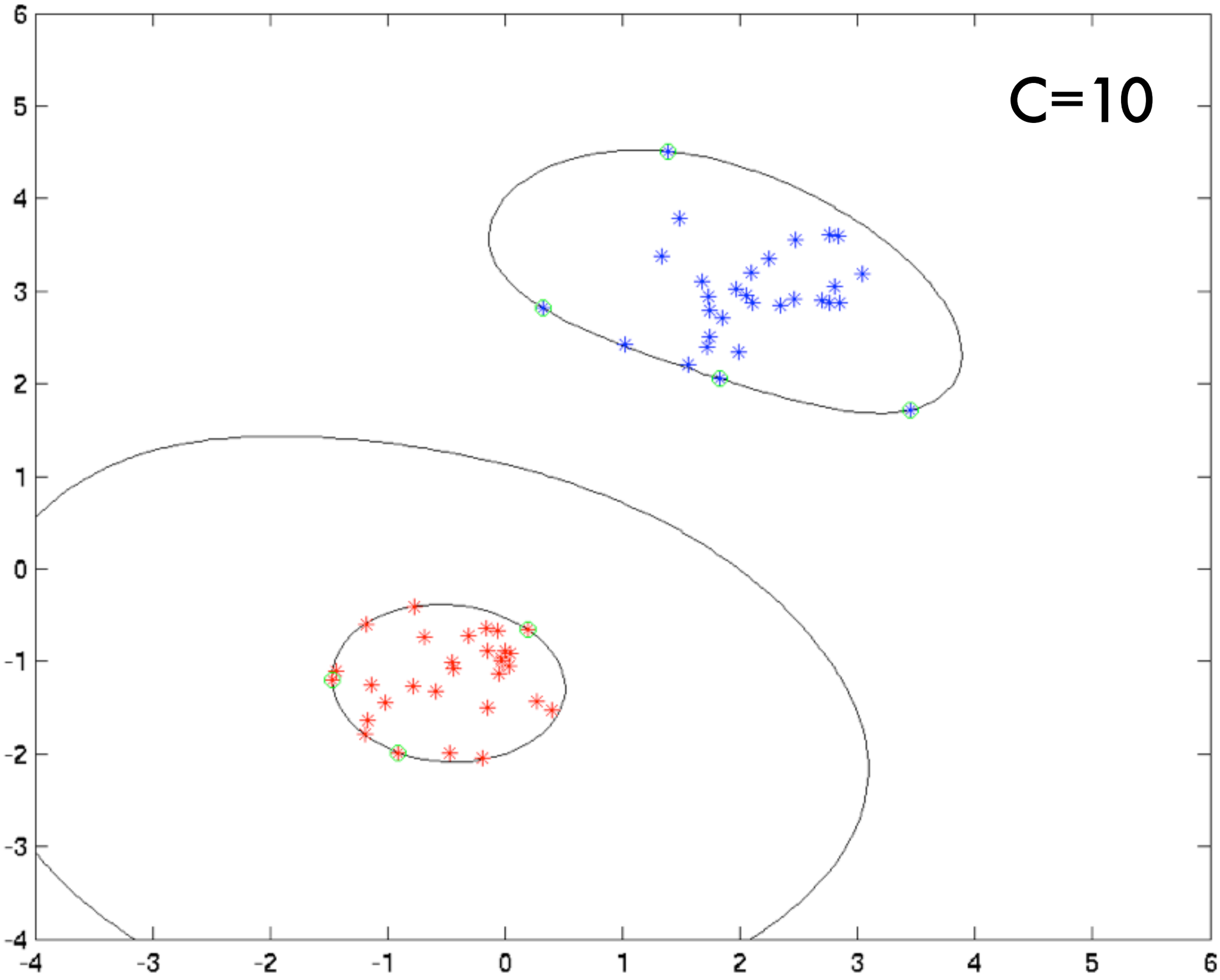

C=50
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

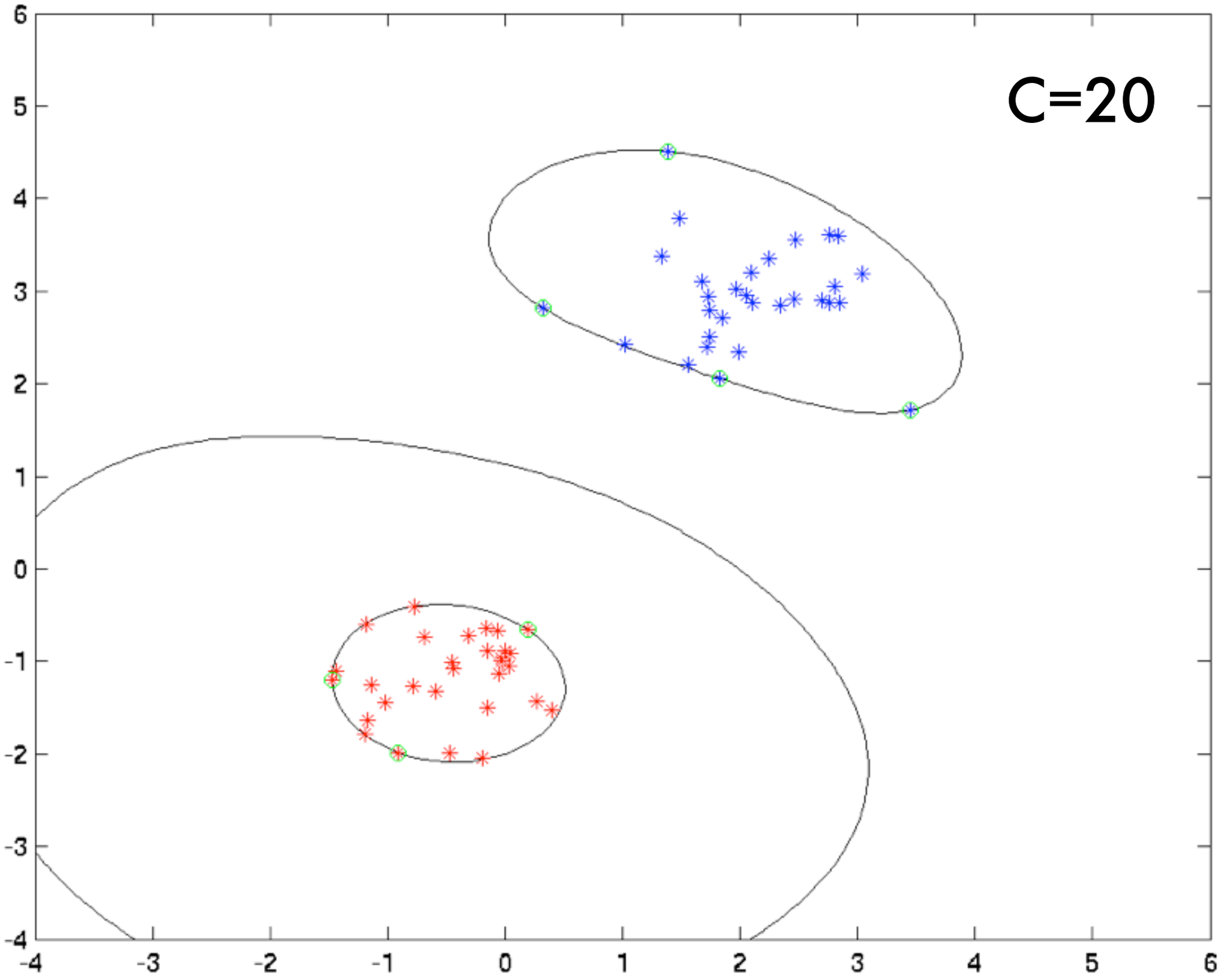

C=100
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

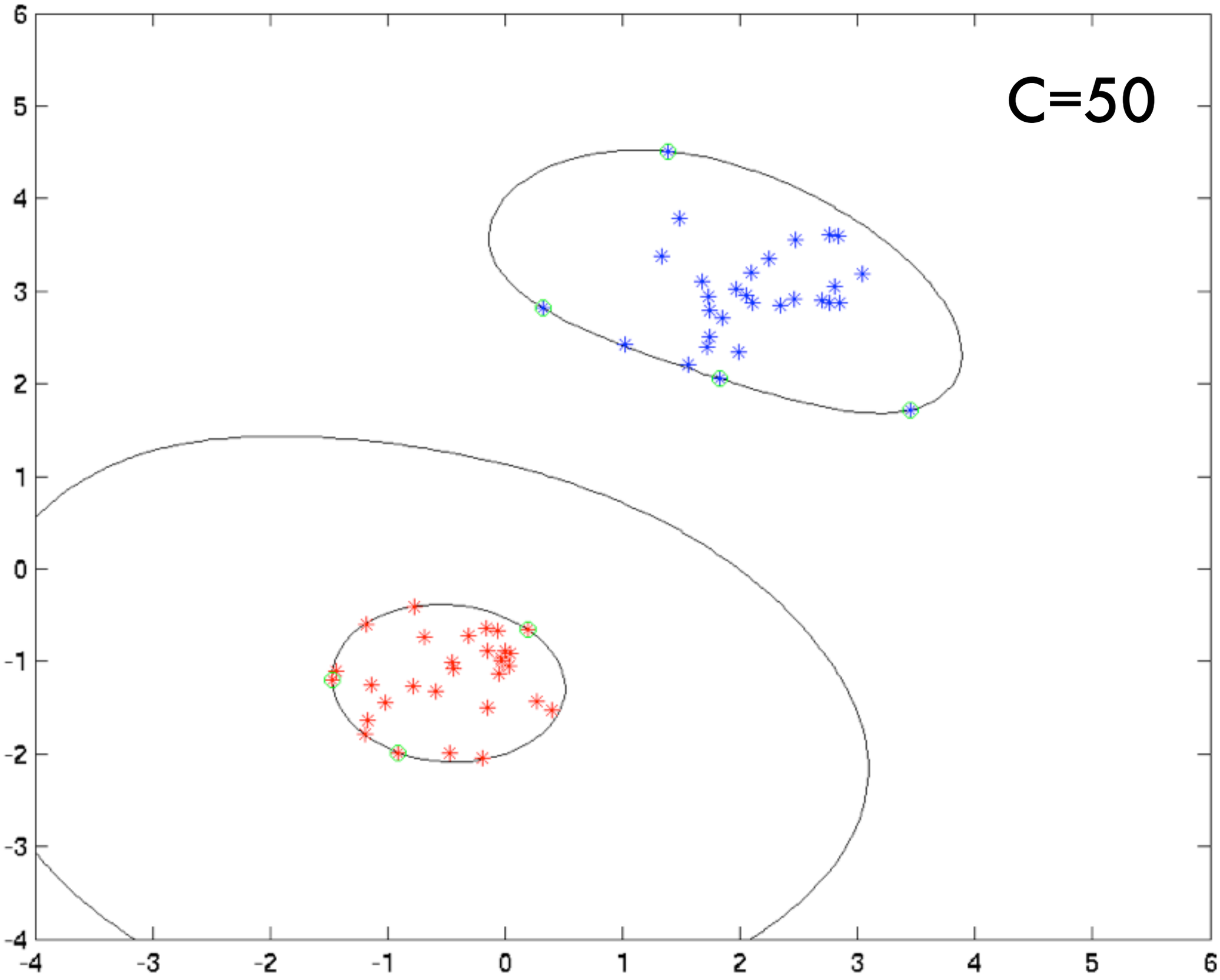

C=1
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

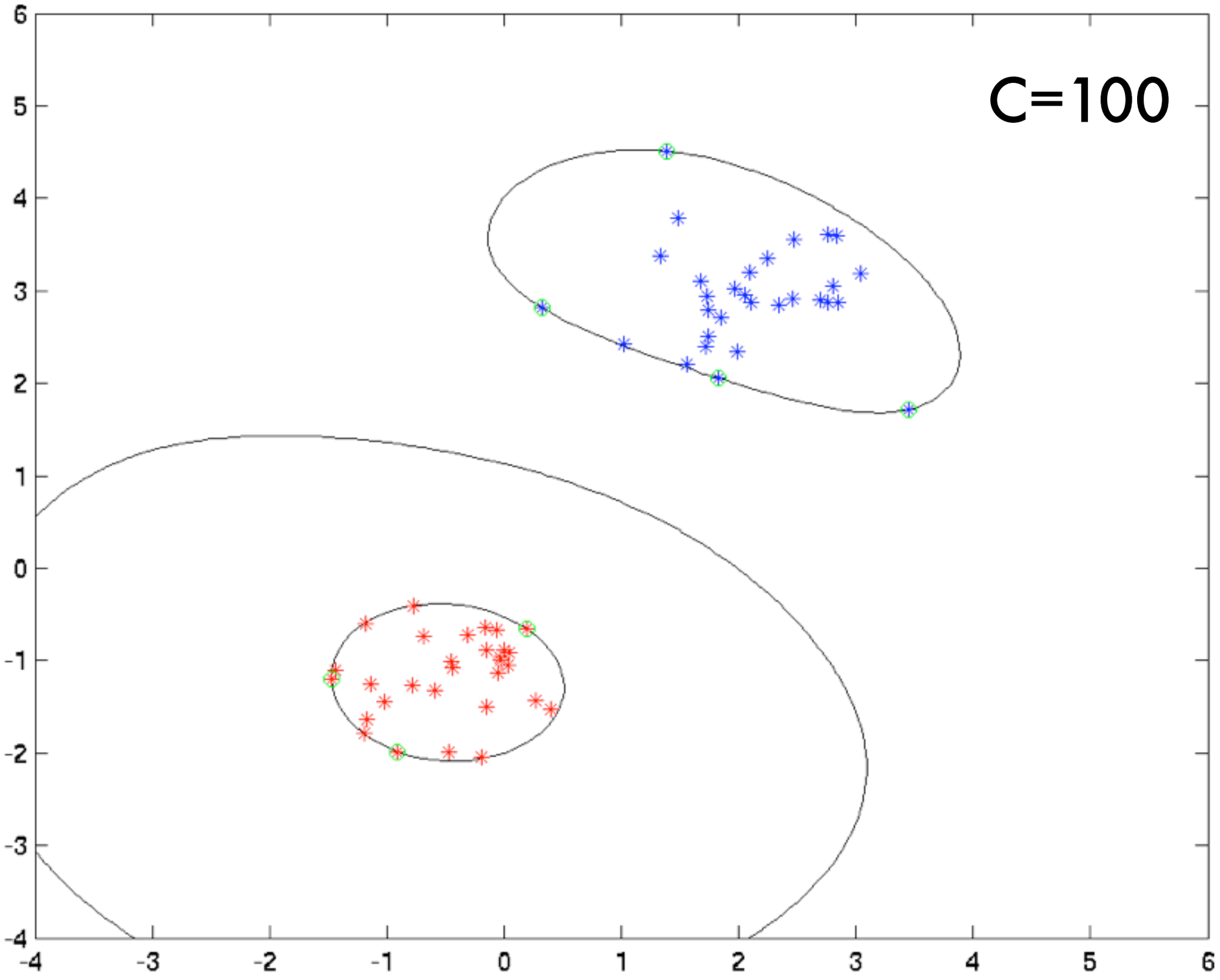

C=2
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

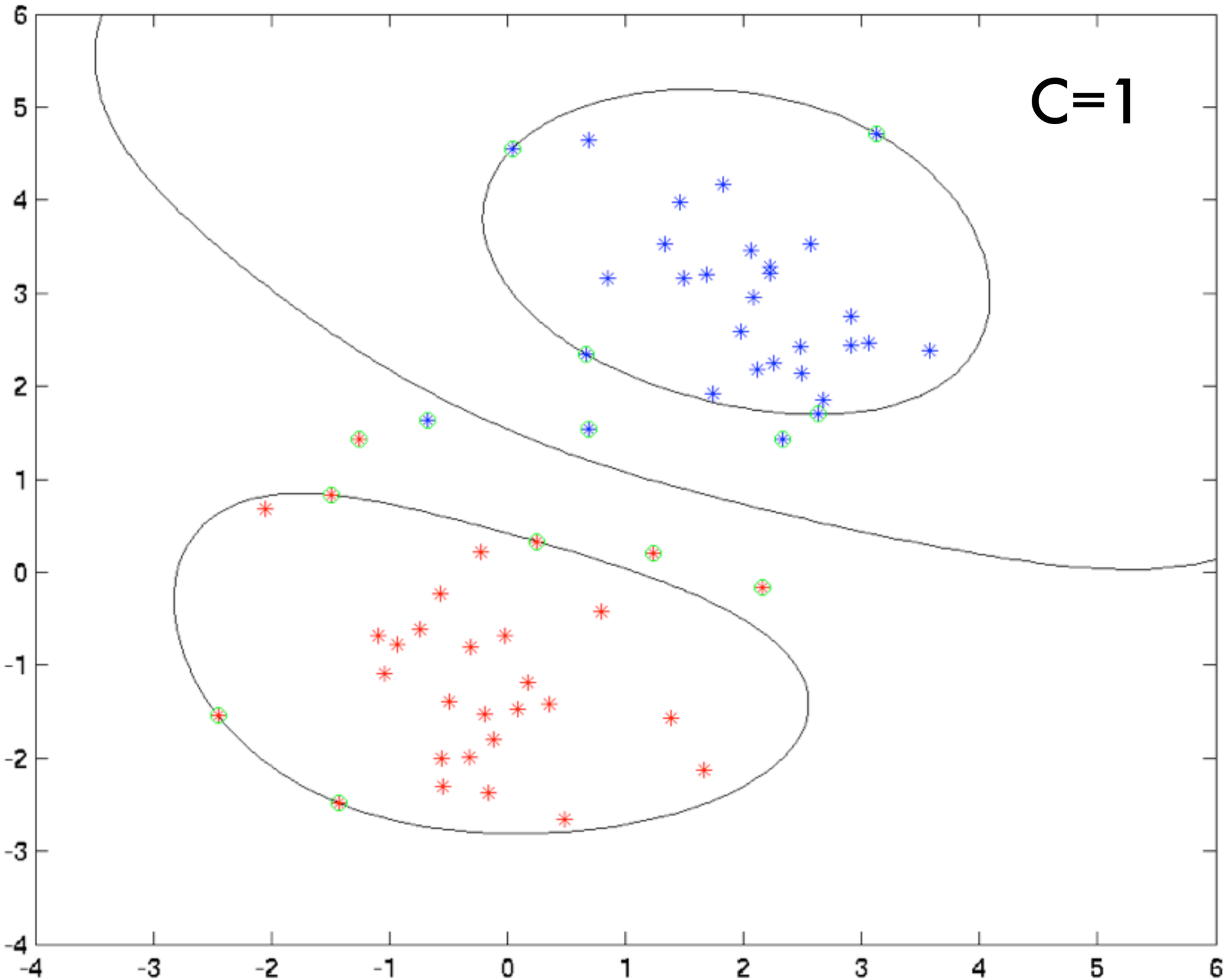

C=5
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

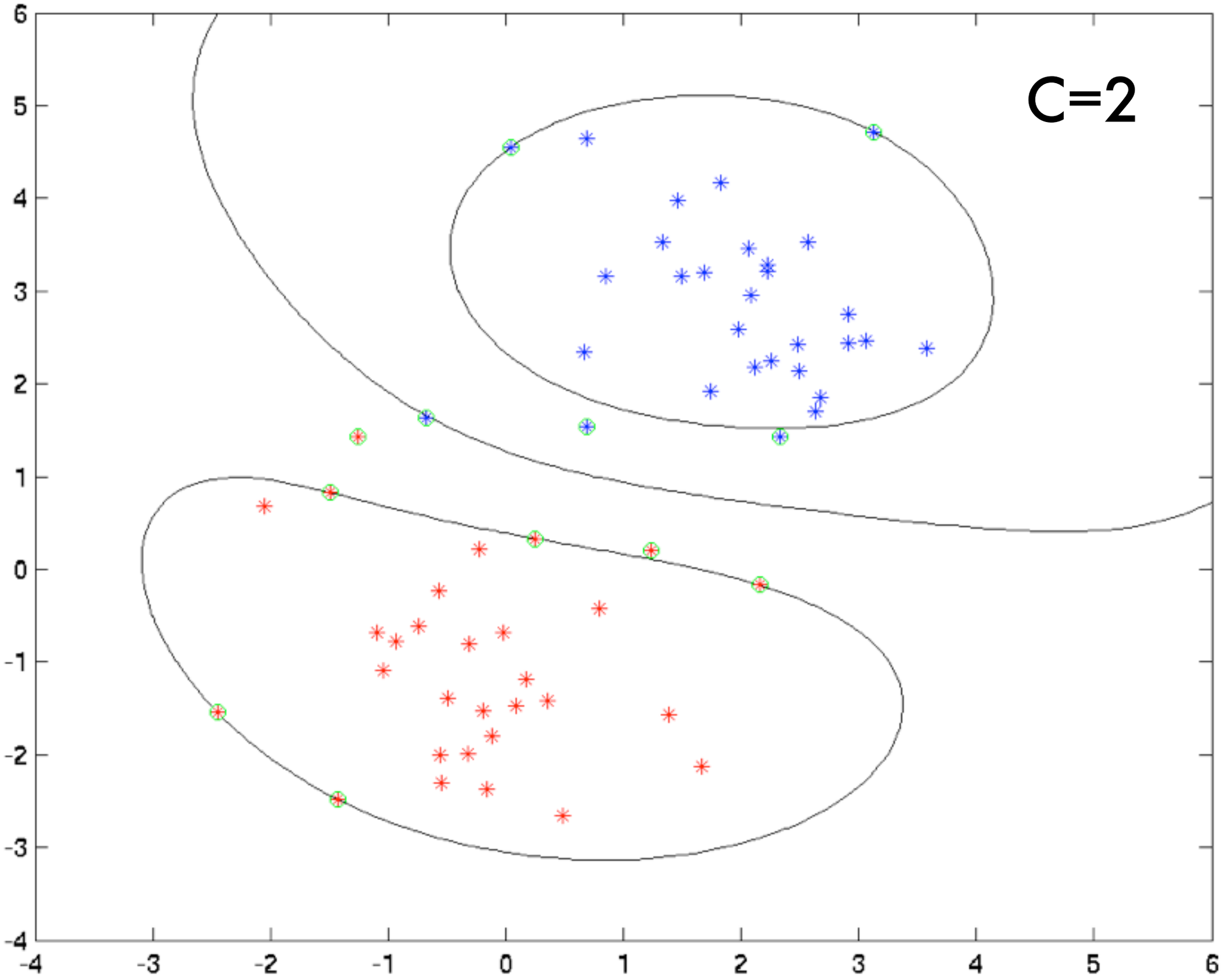

C=10
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

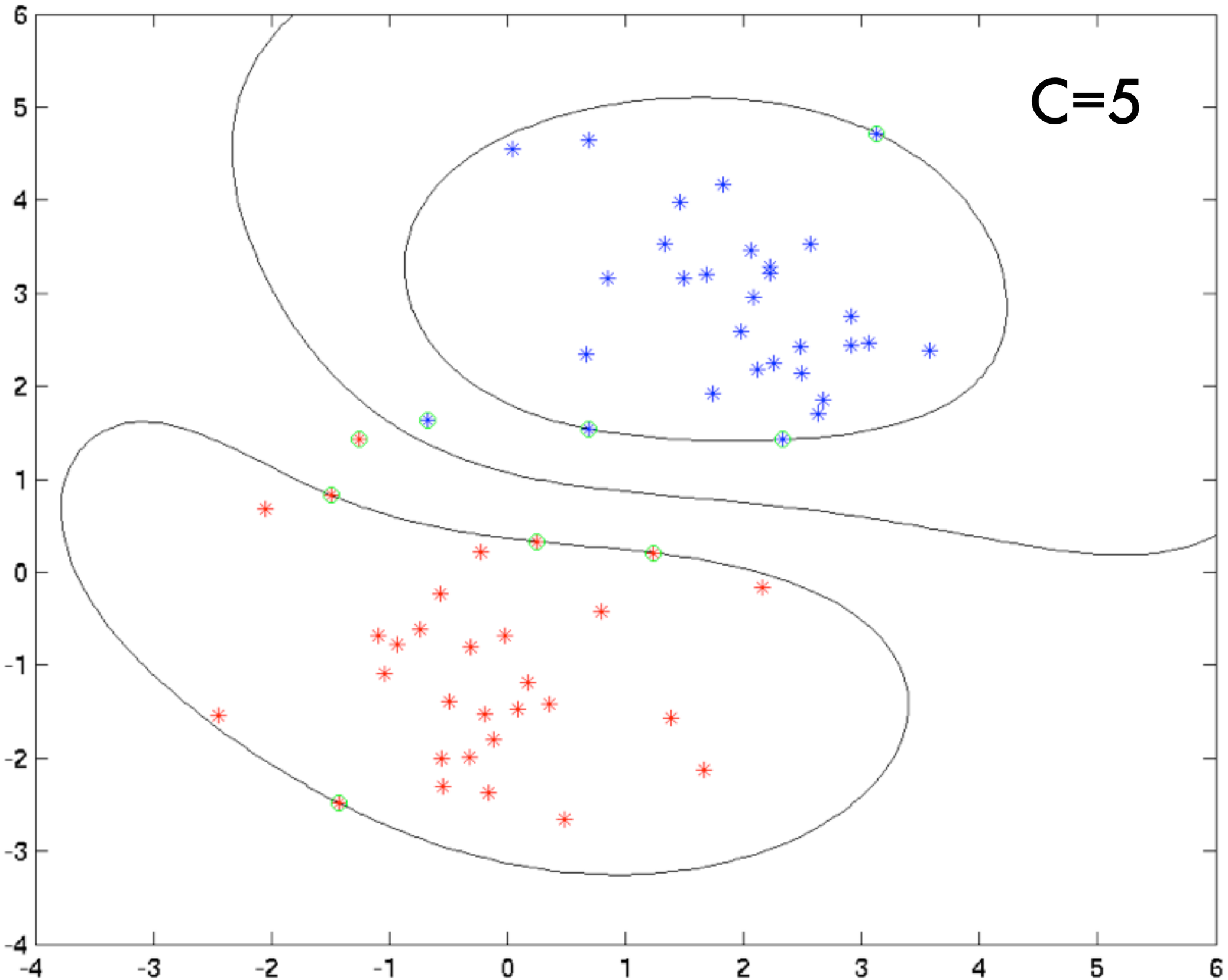

C=20
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

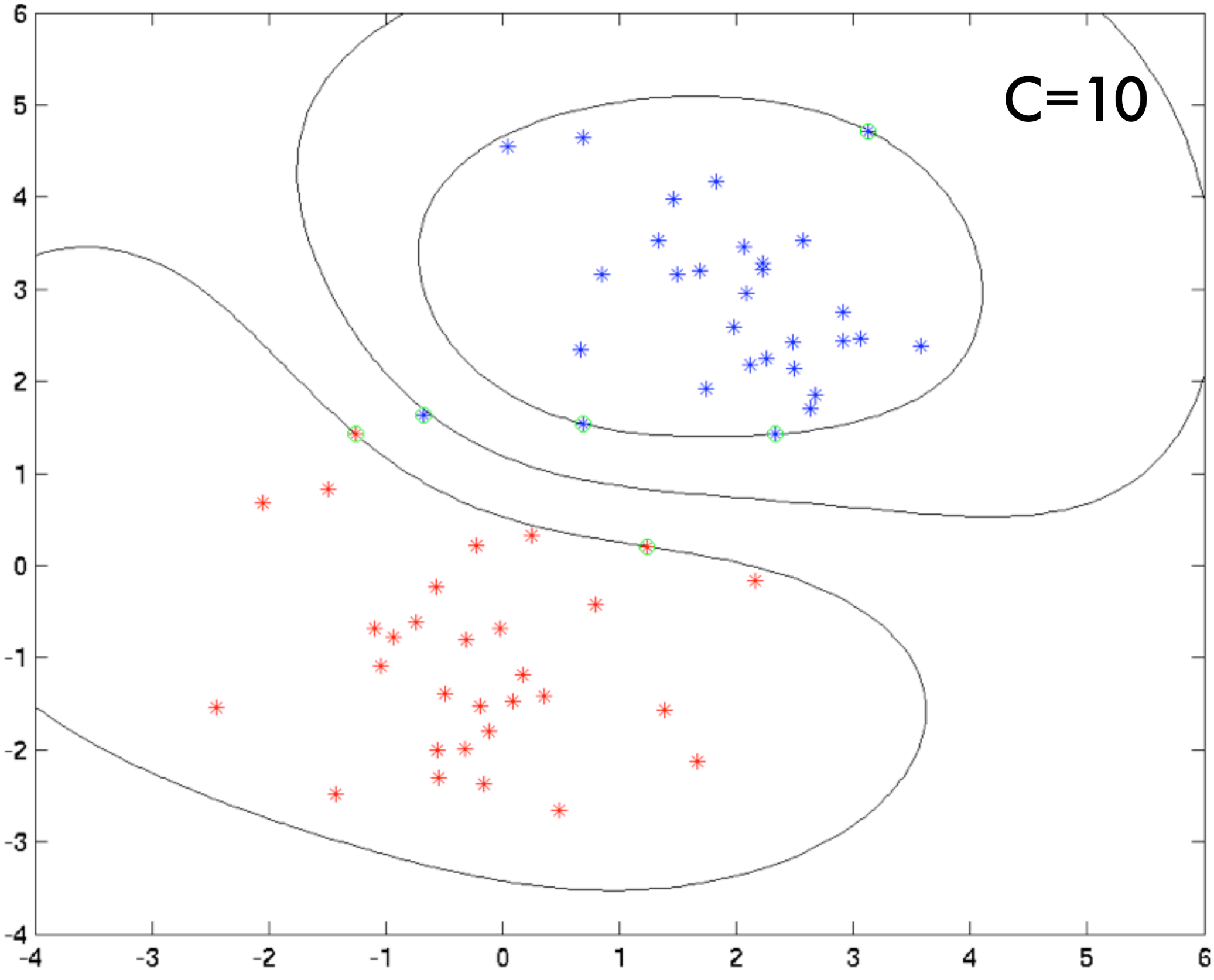

C=50
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

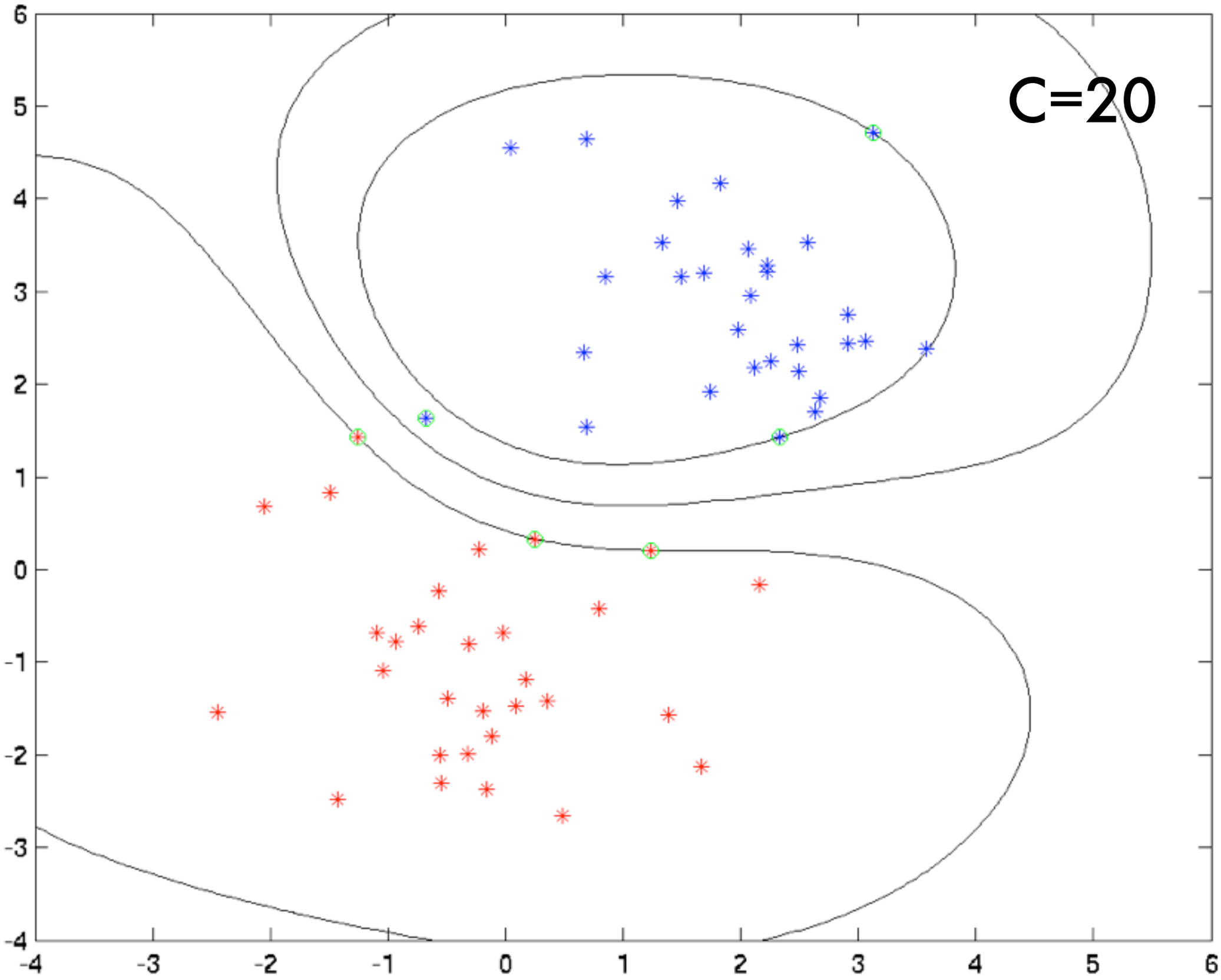

C=100
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

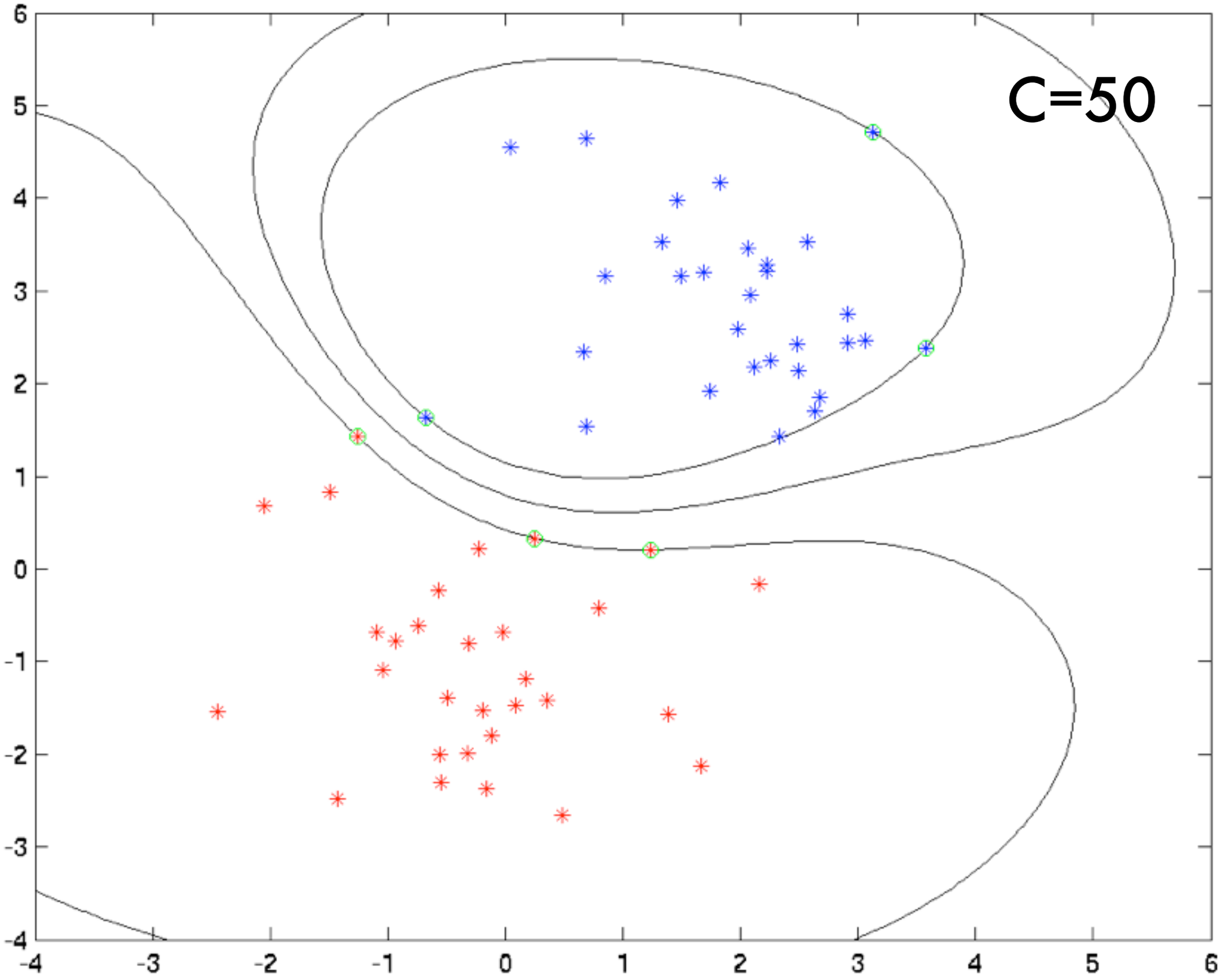

C=1
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

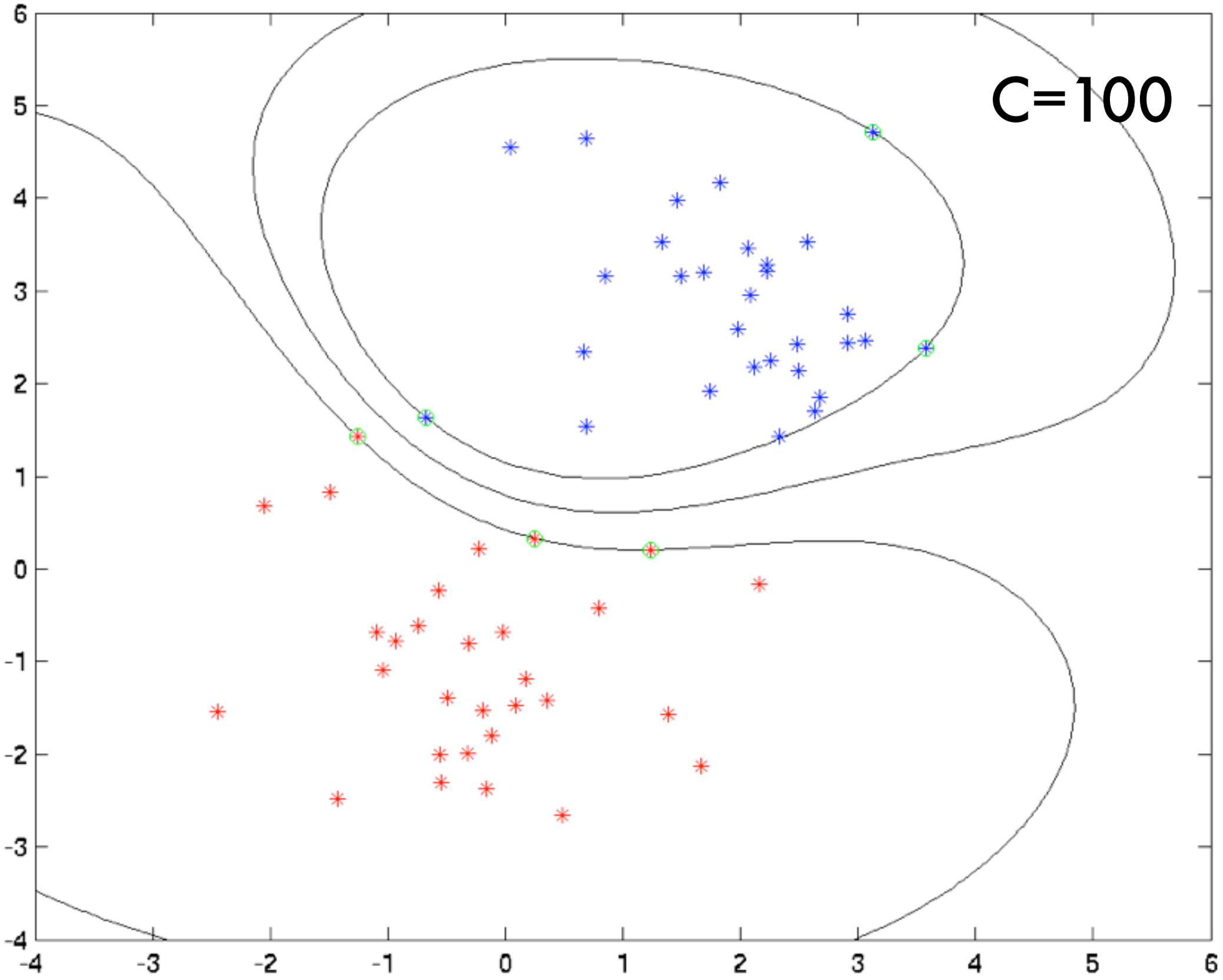

C=2
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

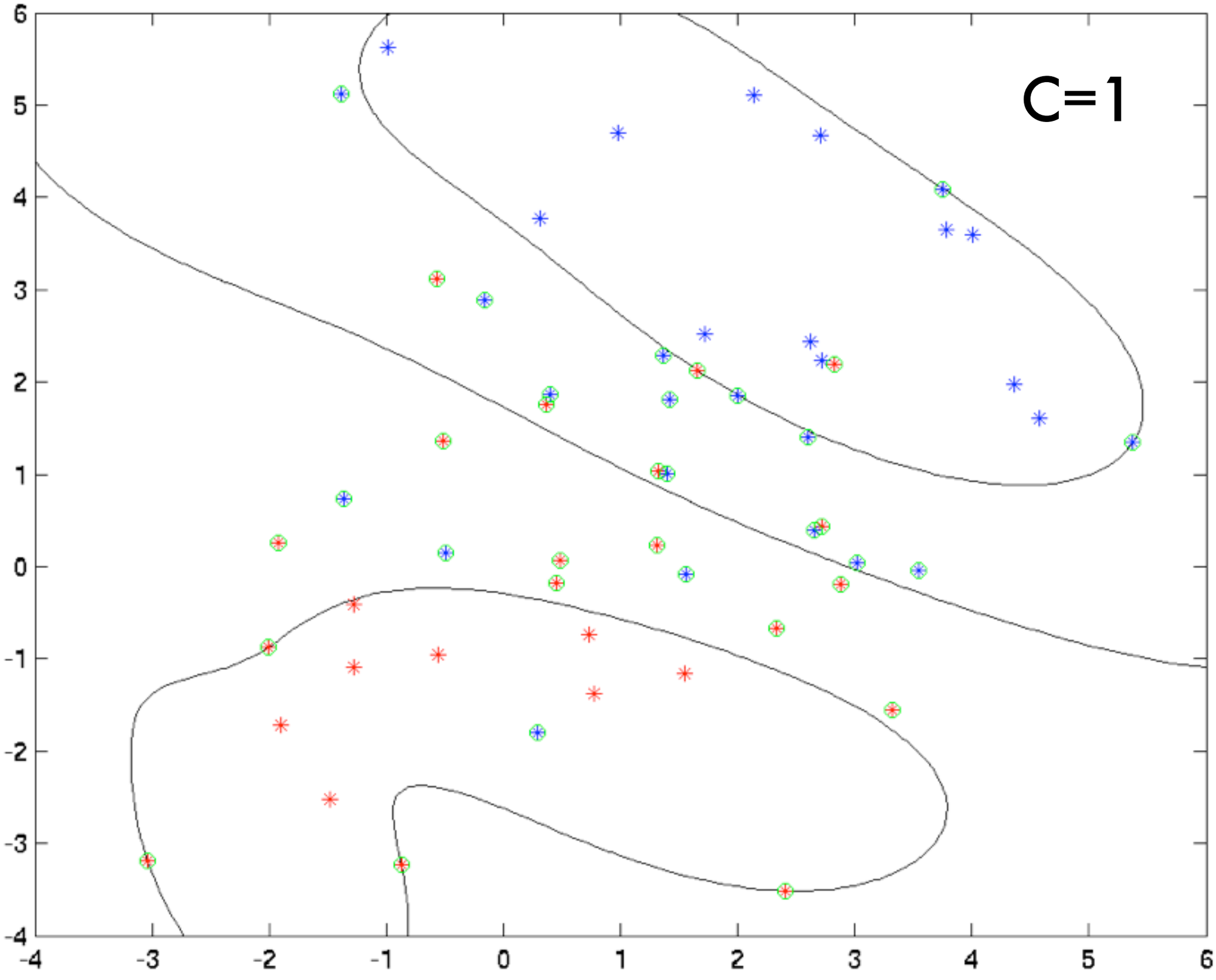

C=5
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

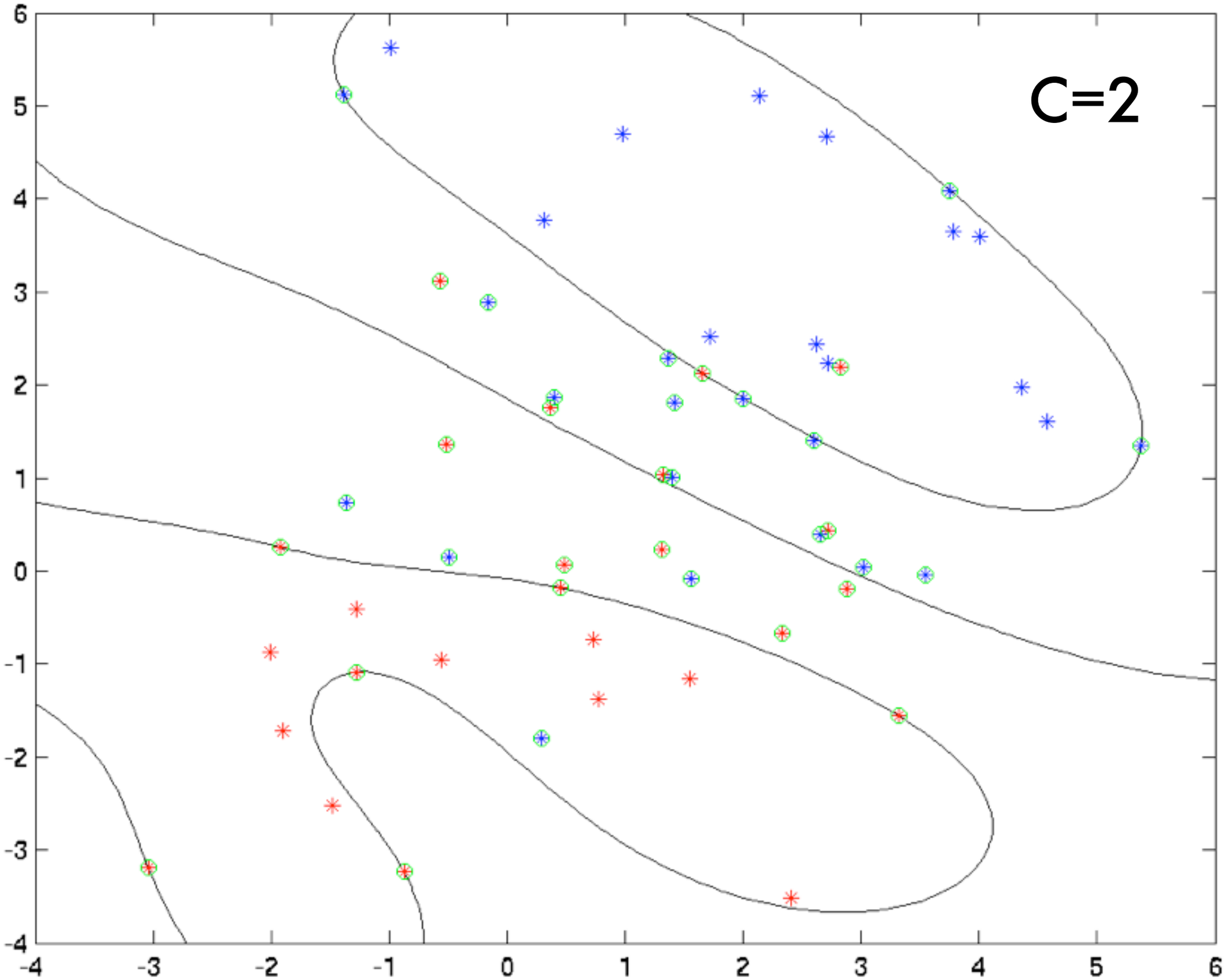

C=10
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

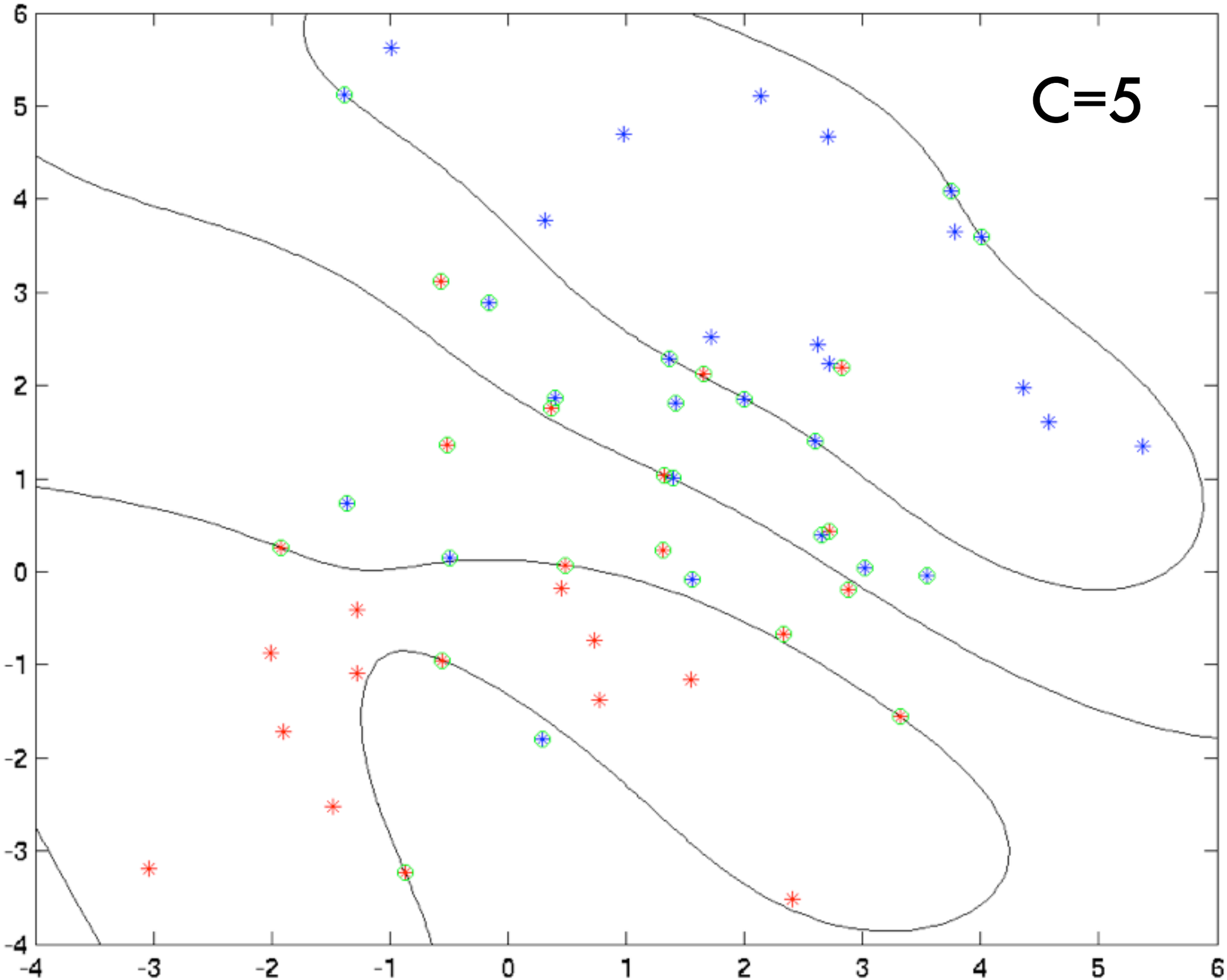

C=20
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

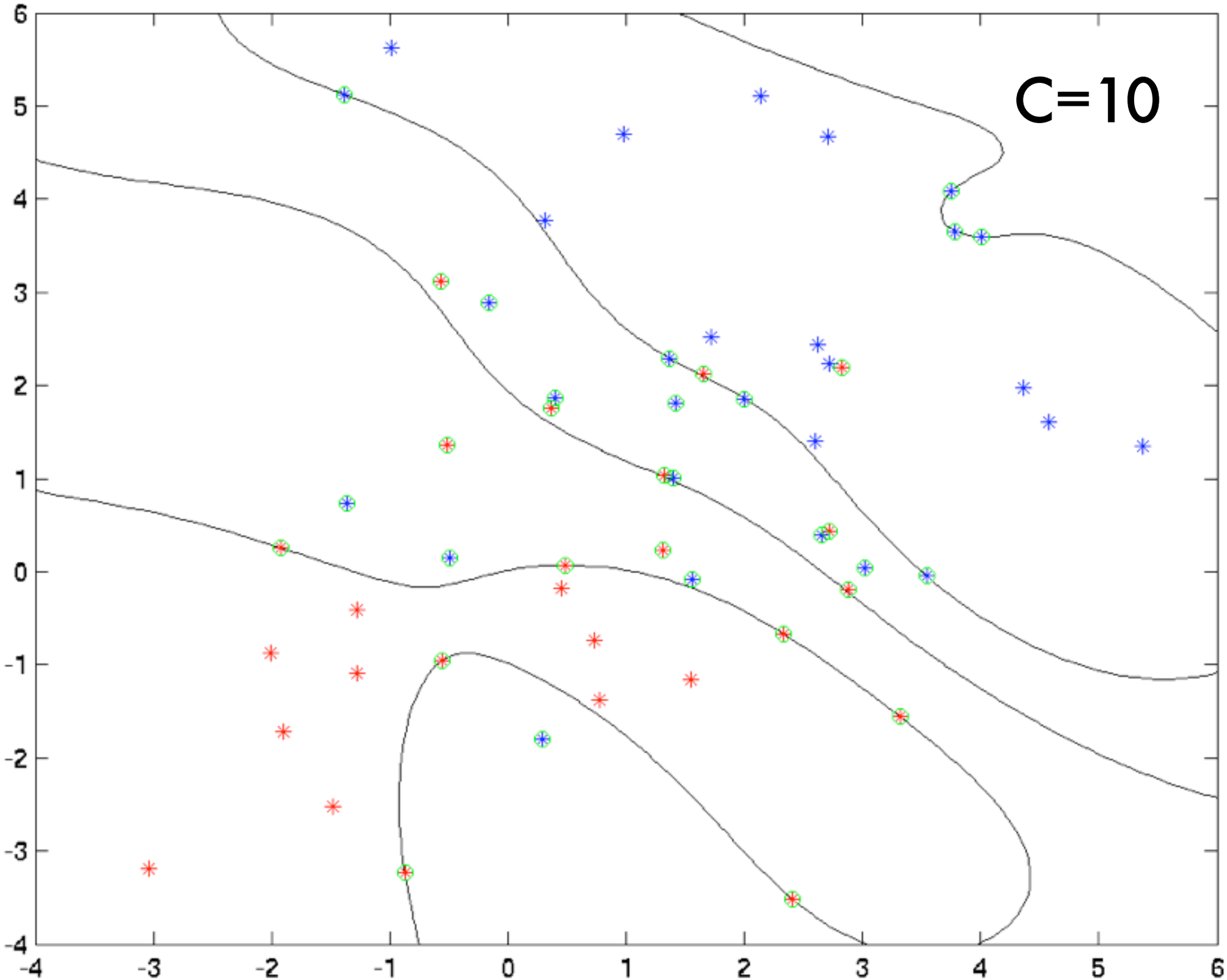

C=50
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

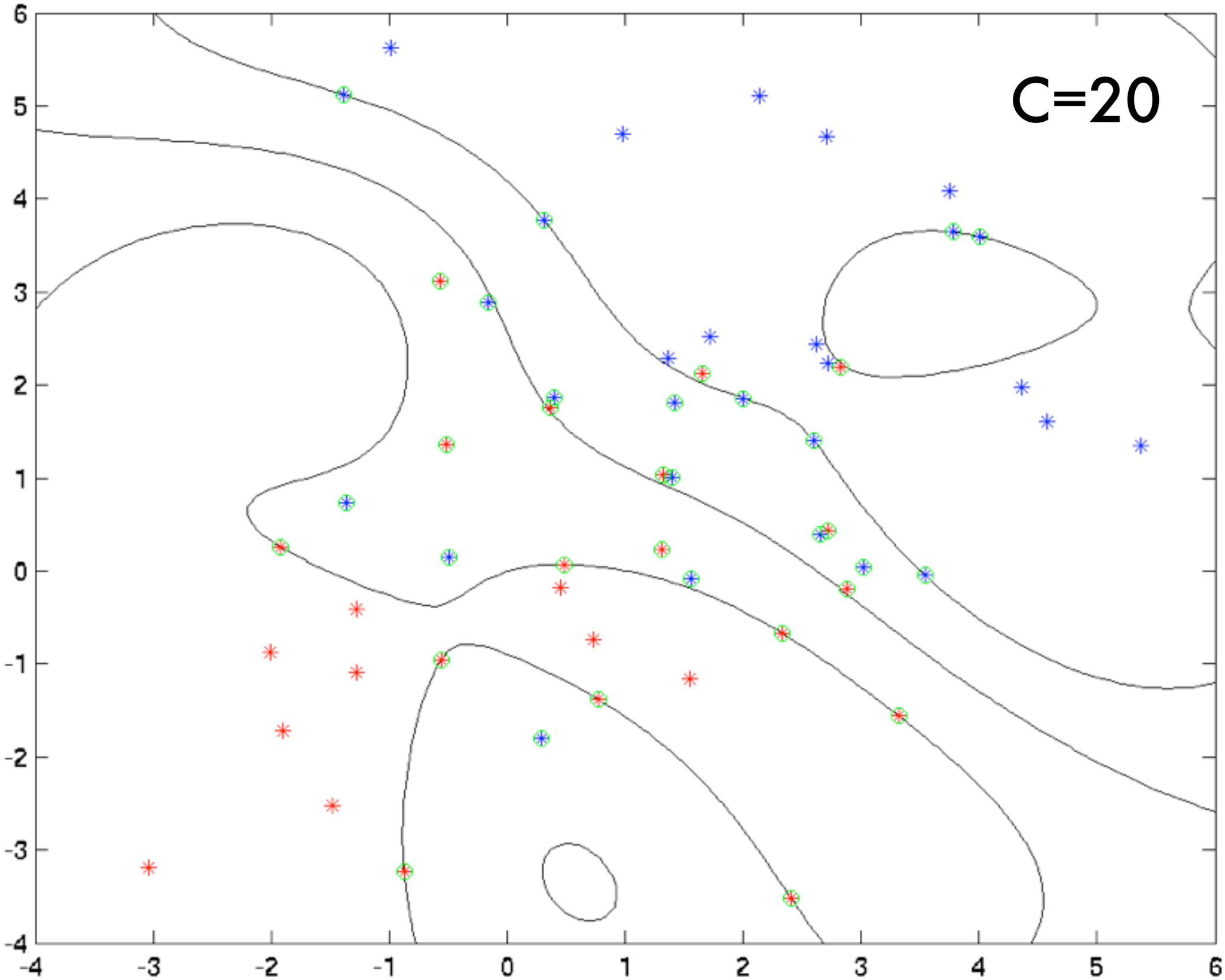

C=100
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

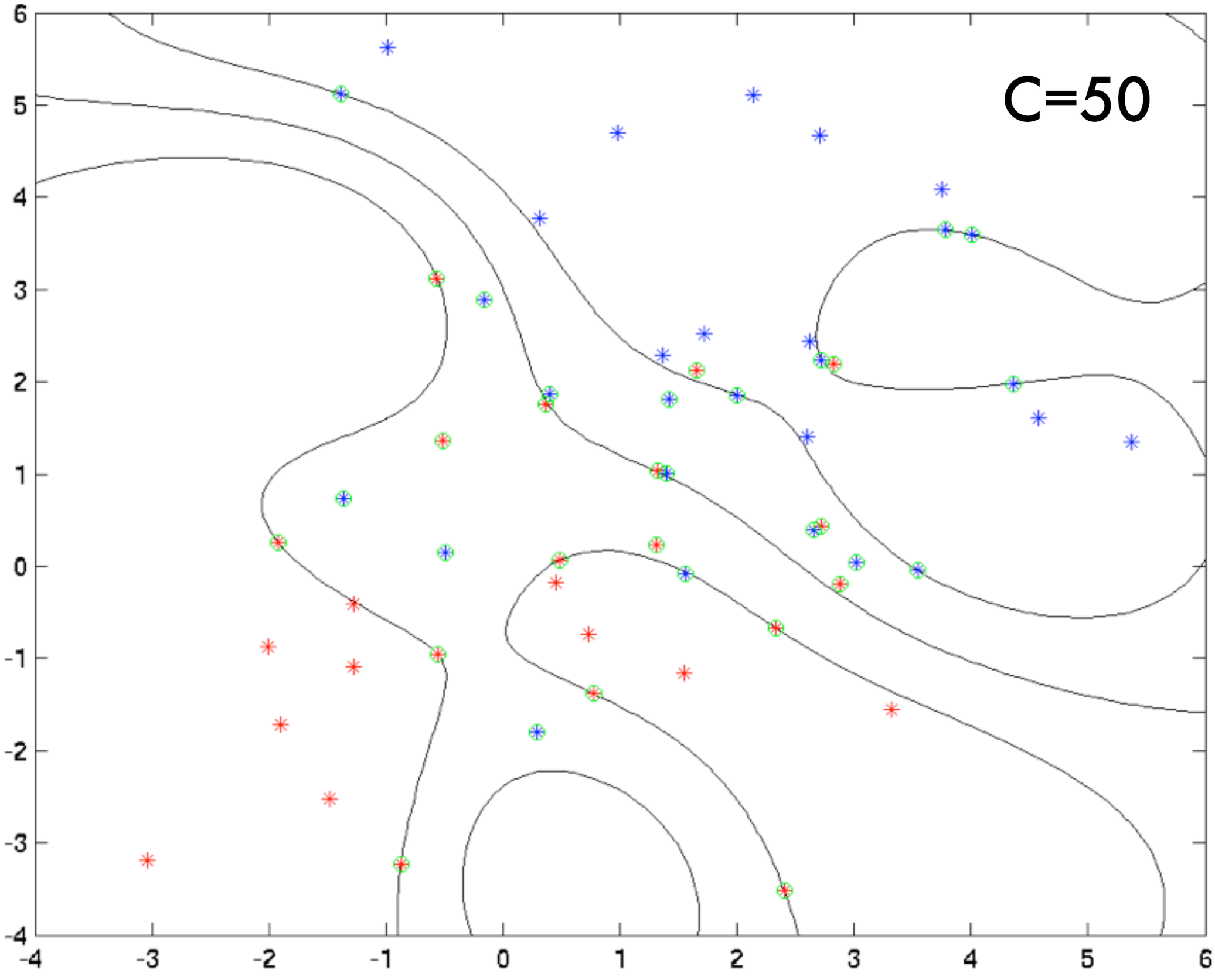

C=1
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

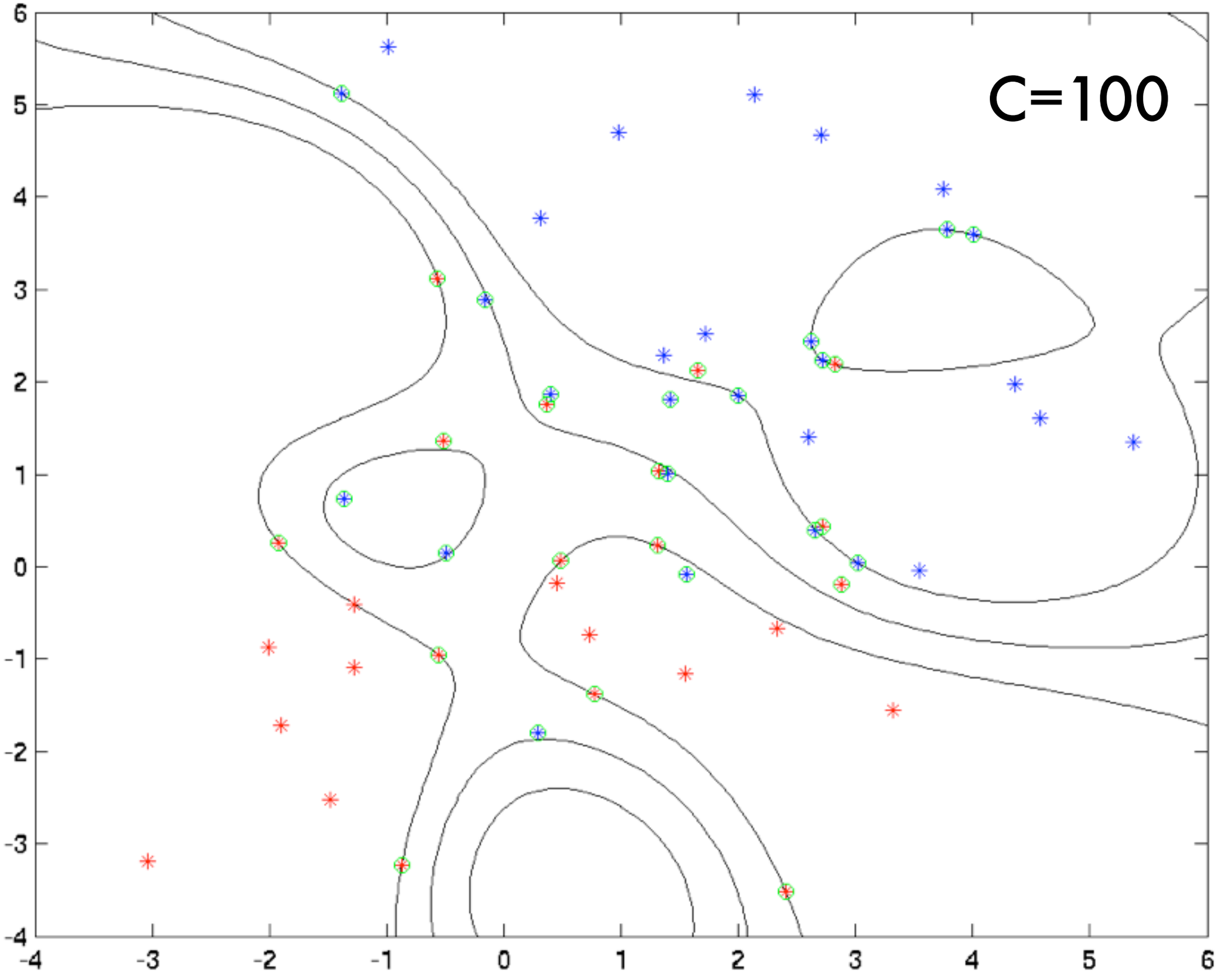

C=2
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6


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

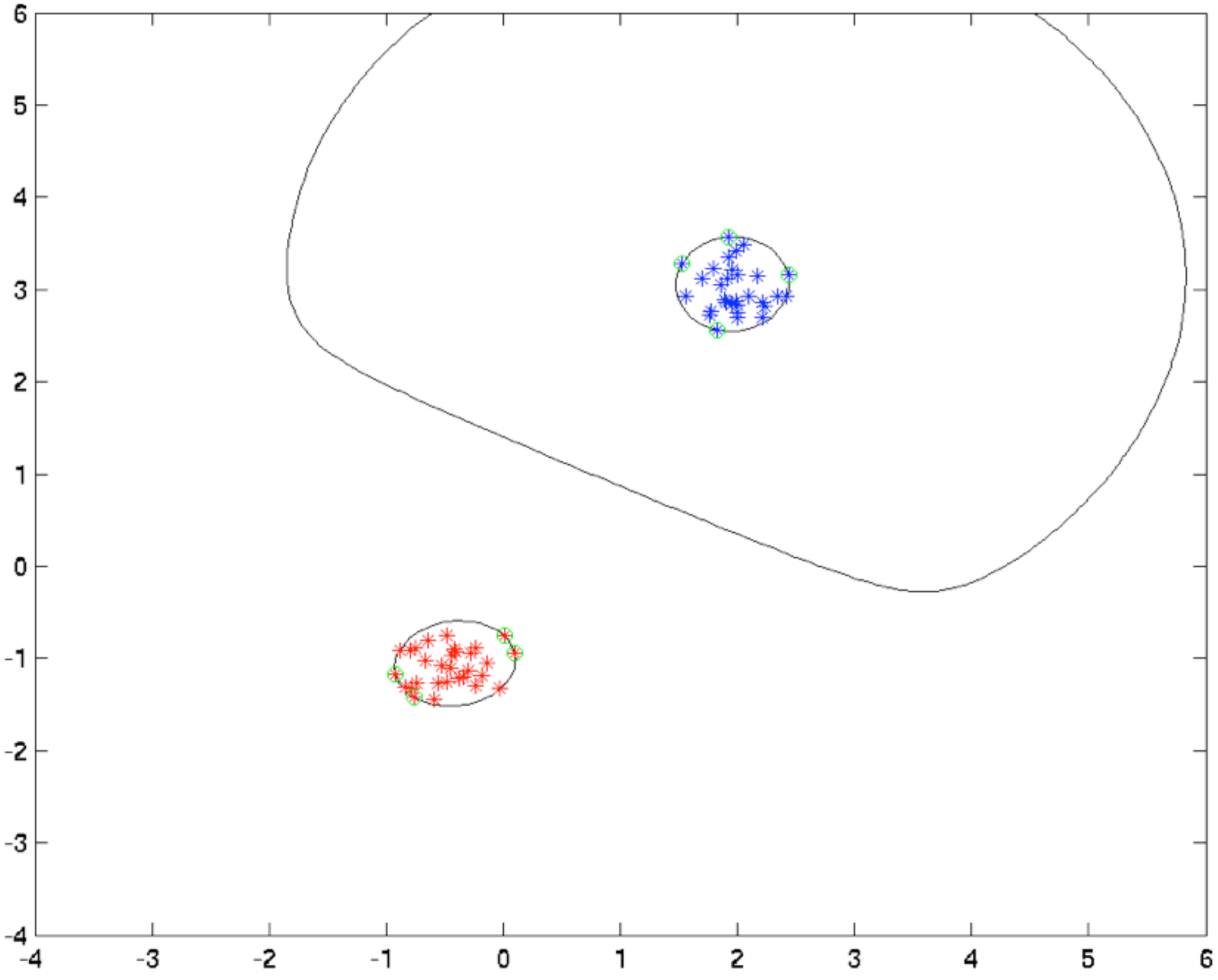

C=10
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

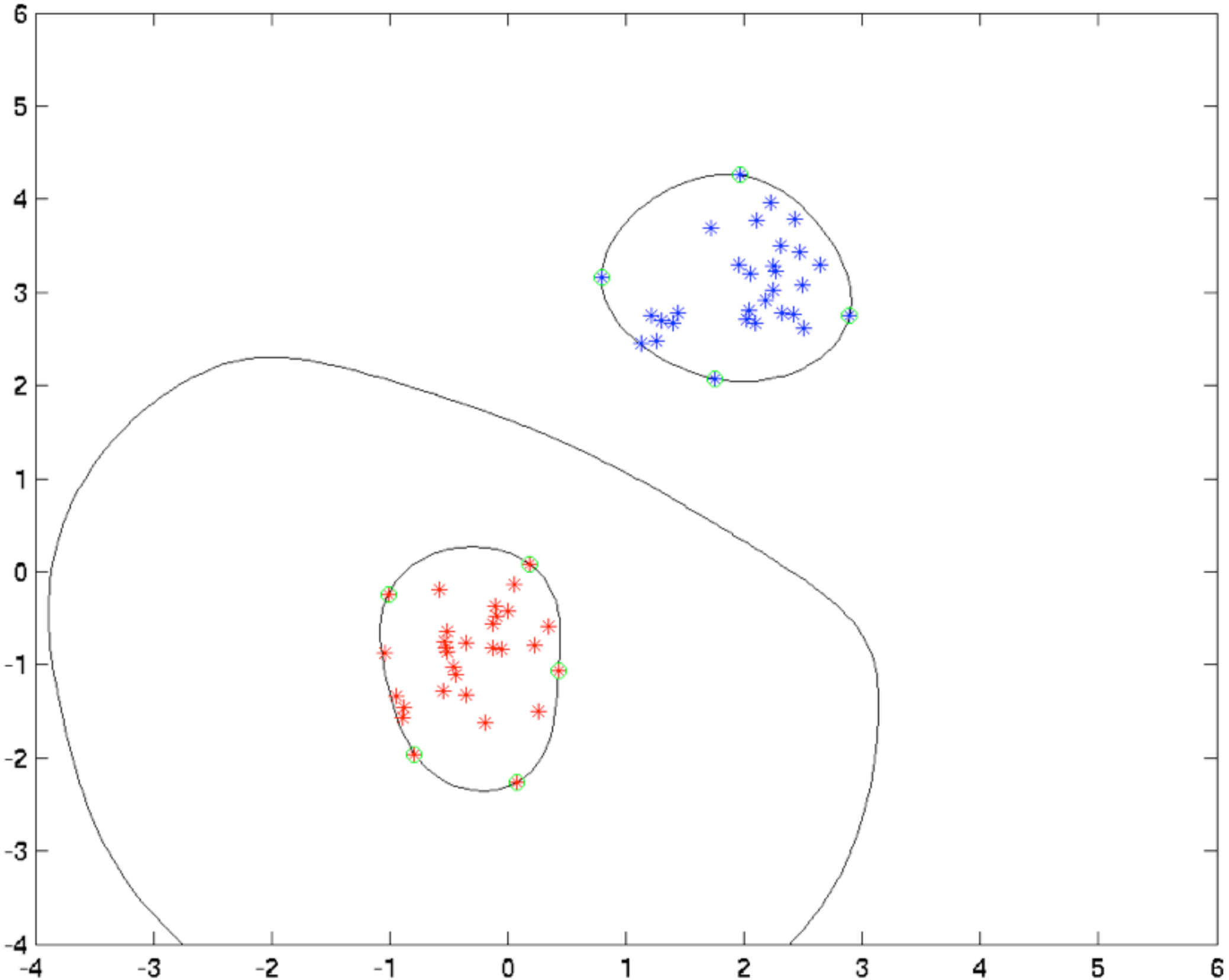

C=20
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

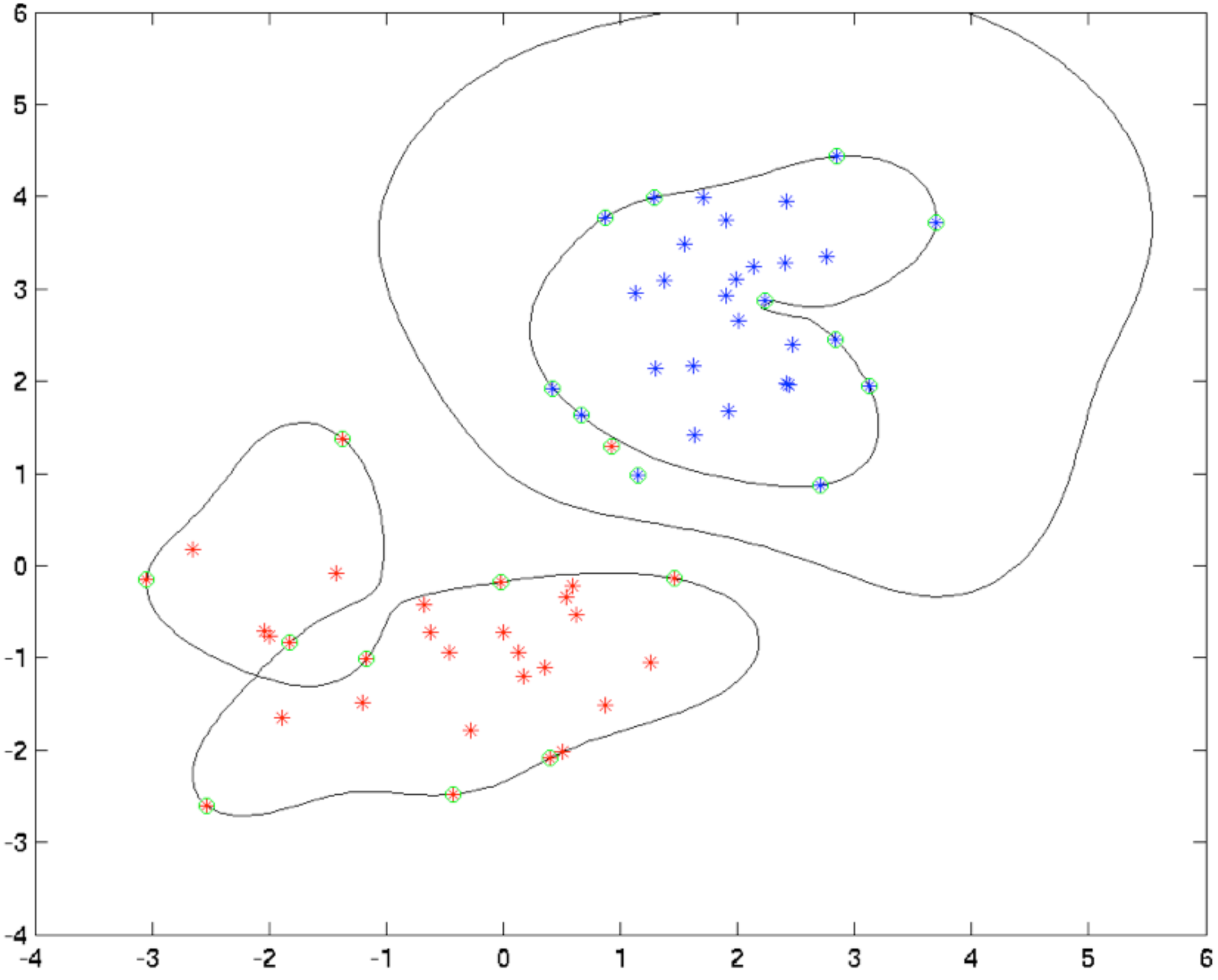

C=50
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

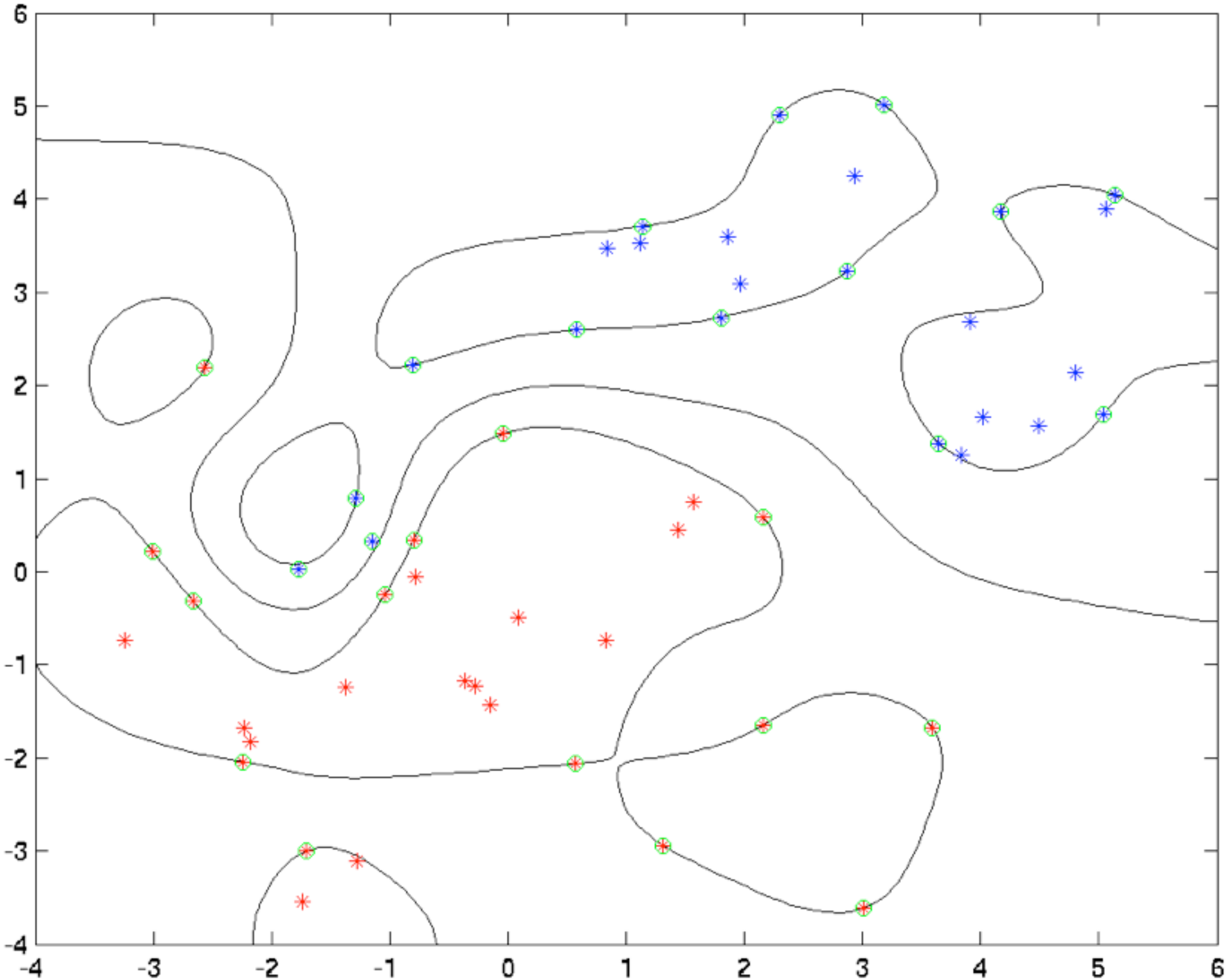

C=100
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

And now with a narrower kernel

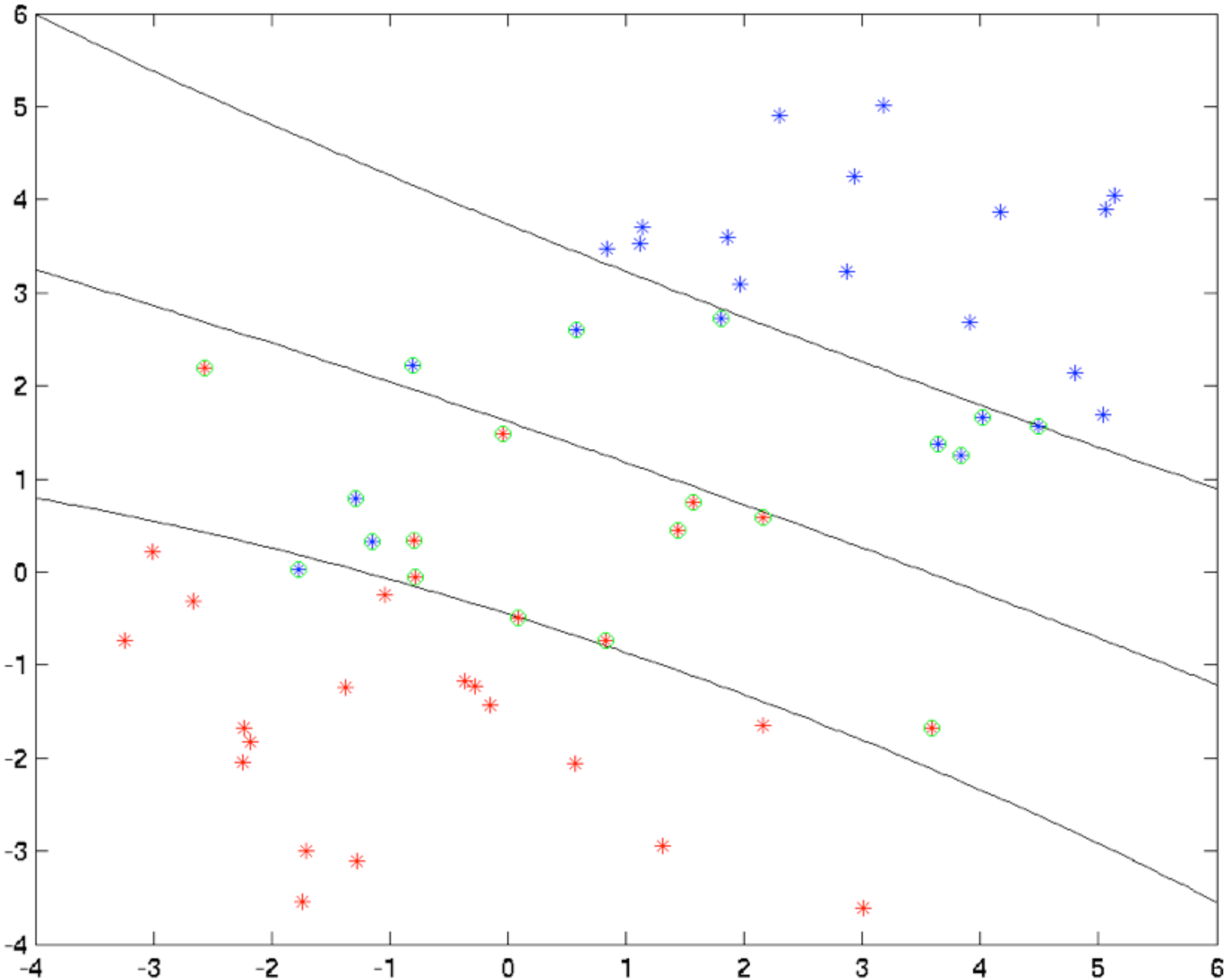

6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6


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

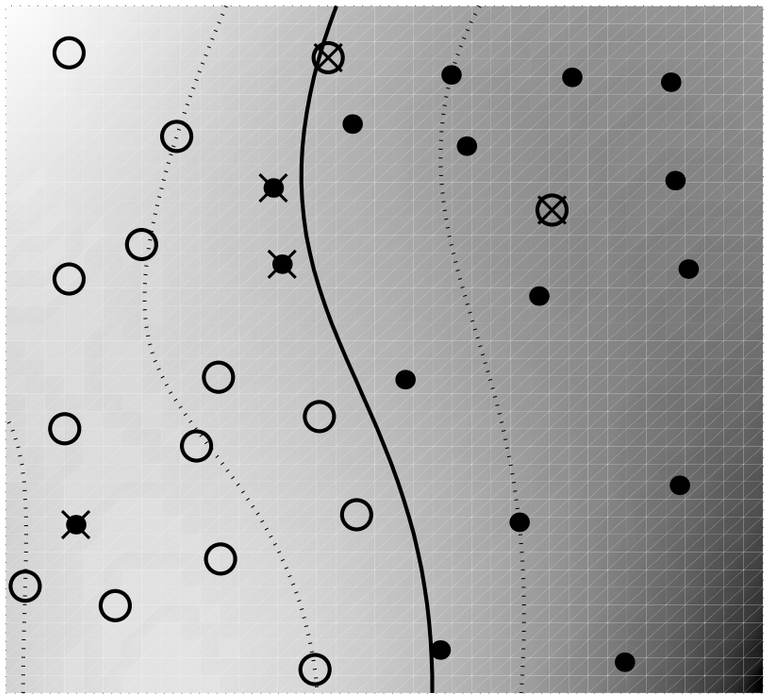

6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

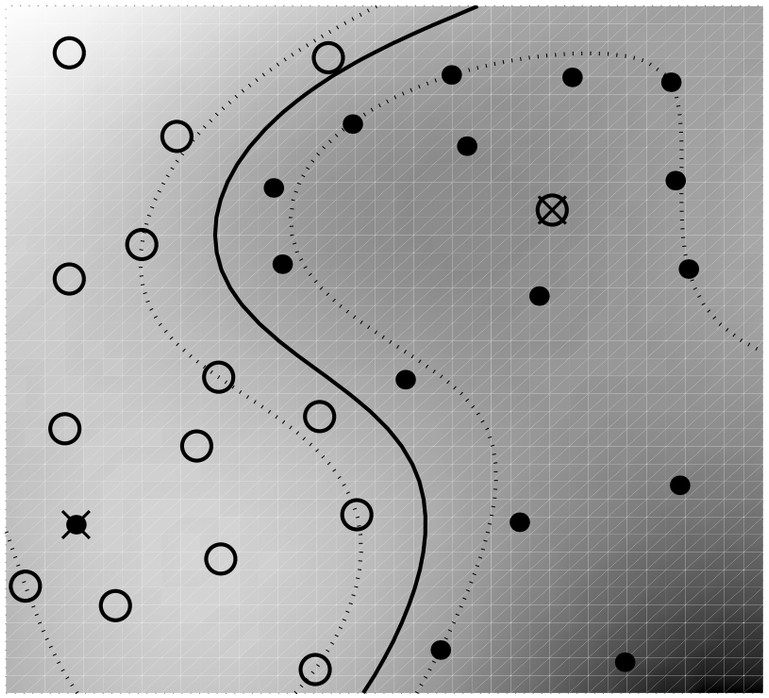

6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

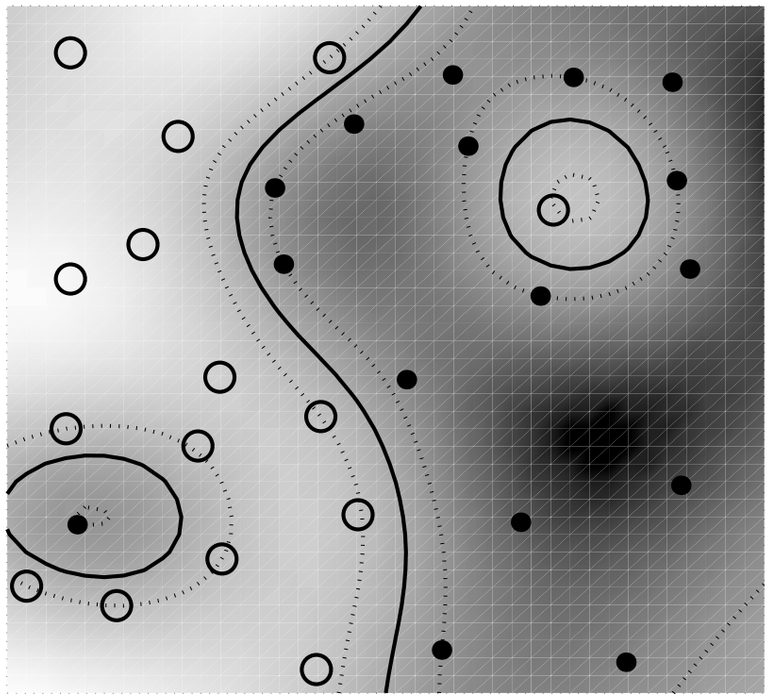
And now with a very wide kernel


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

Nonlinear separation







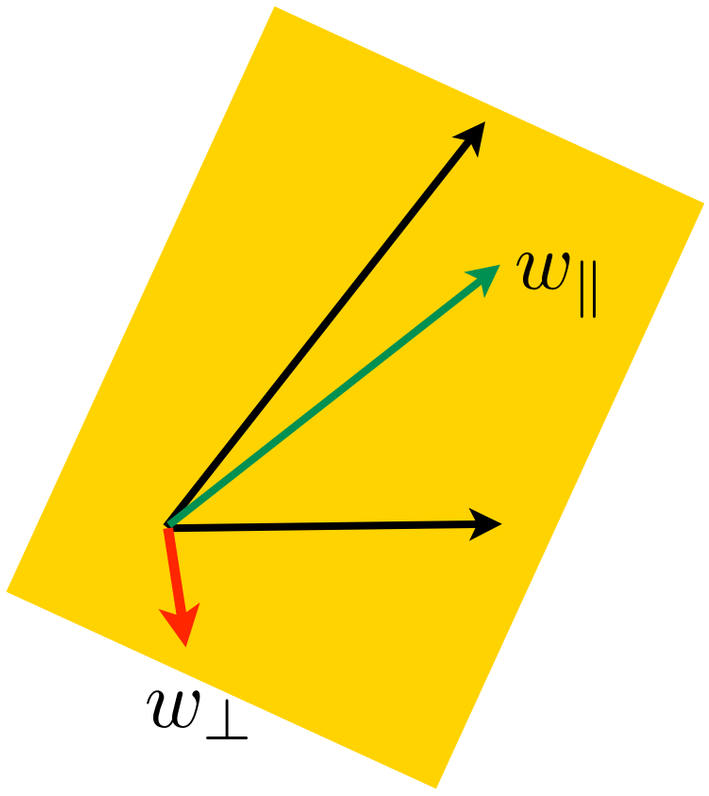
- Increasing C allows for more nonlinearities
- Decreases number of errors
- SV boundary need not be contiguous
- Kernel width adjusts function class

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)



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

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 $w = \sum_i \alpha_i \phi(x_i)$
  - 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 $\alpha = (K + m\lambda \mathbf{1})^{-1} y$ as linear system

Many more applications

- (Kernel) Principal Component Analysis
- (Kernel) Independent Component Analysis
- (Kernel) Linear Discriminant Analysis
- (Kernel) Canonical Correlation Analysis
- (Kernel) Two-Sample Test
- (Kernel) Graphical Models
- (Kernel) Covariate Shift Correction
- (Kernel) …

Outline

- **Convex Optimization**
  - Unconstrained Optimization
  - Constrained Optimization and Duality
  - Linear and Quadratic programs
- **Support Vector Machines**
  - Classification
  - Regression
  - Novelty Detection
- **Kernels**
  - Feature Space
  - Kernel PCA
  - Kernelized SVM