6.867 Fall 2017
Introduction to Classification

Support Vector Machines

Lecture 7: 28th Sept., 2017

Admin

No recitation this Friday (student holiday)

Exercises 3 are on Stellar; please work through

Project grouping: **Critical, meet here!**

Milestone 0 is over

Read the HW1 announcement carefully!

Outline

★ Last 'linear' lecture

★ Support Vector Machines (SVMs)

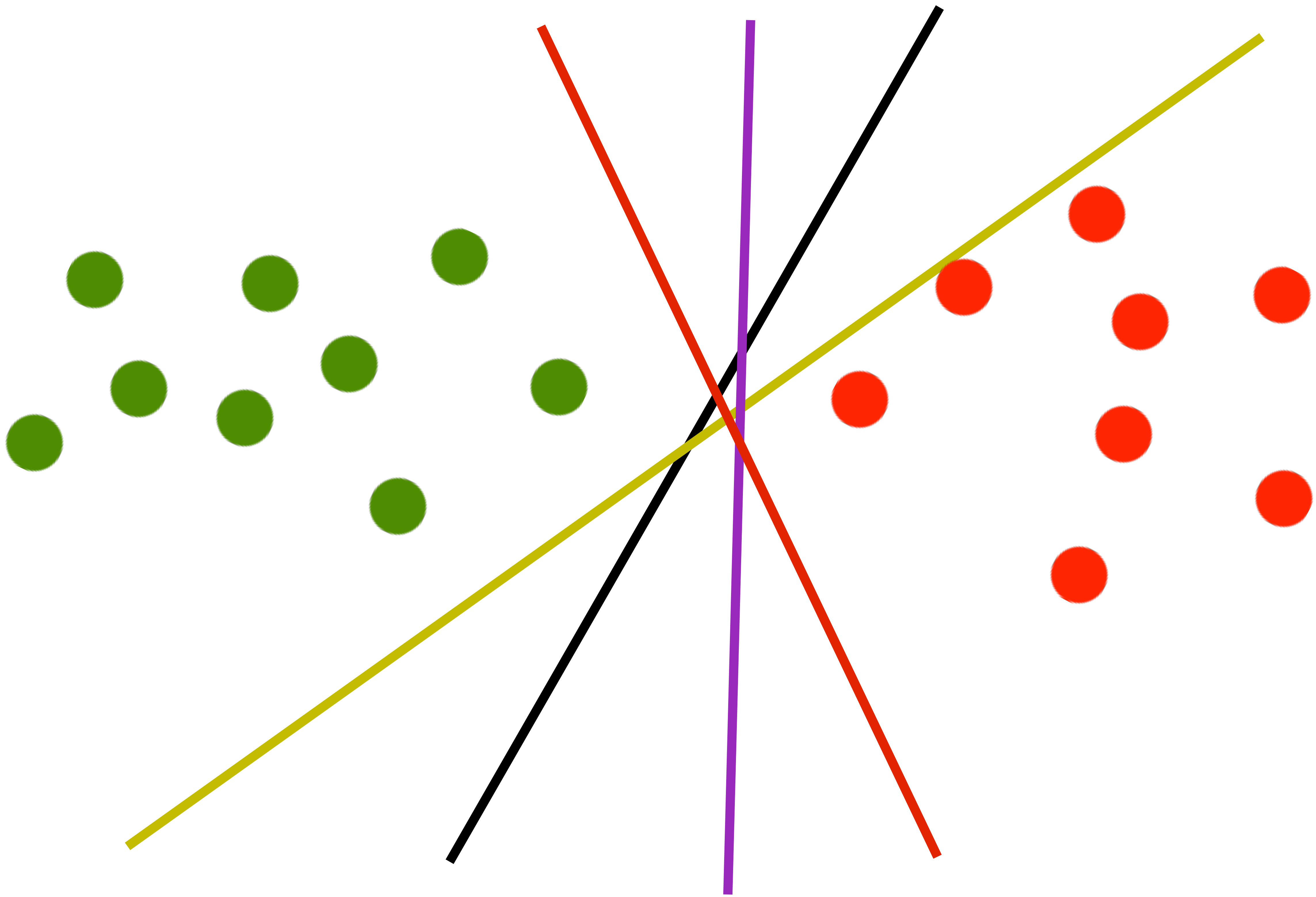
  ★ The notion of margin

  ★ Some history

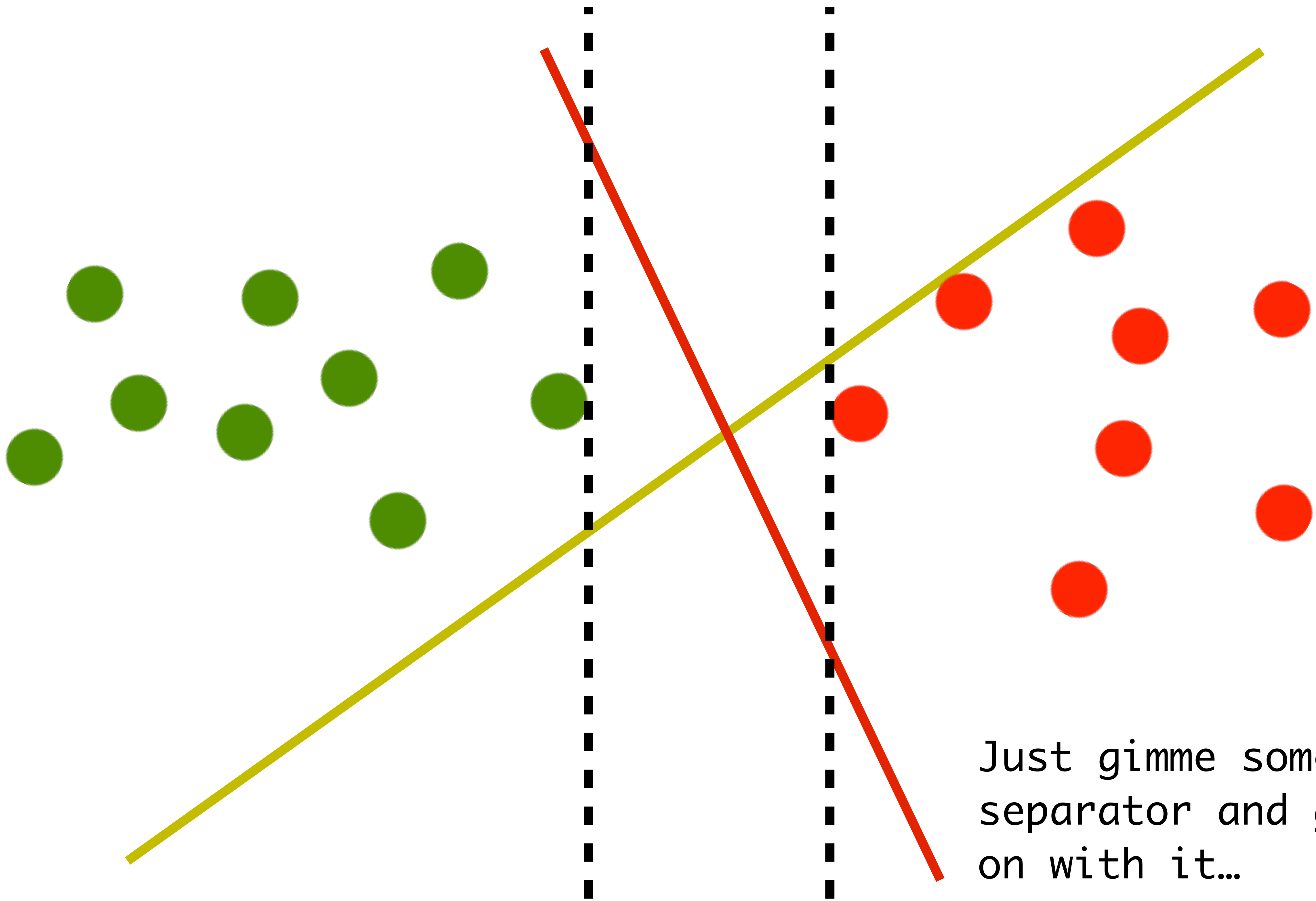
  ★ SVMs and optimization

★ High-level remarks

Linear separators of data

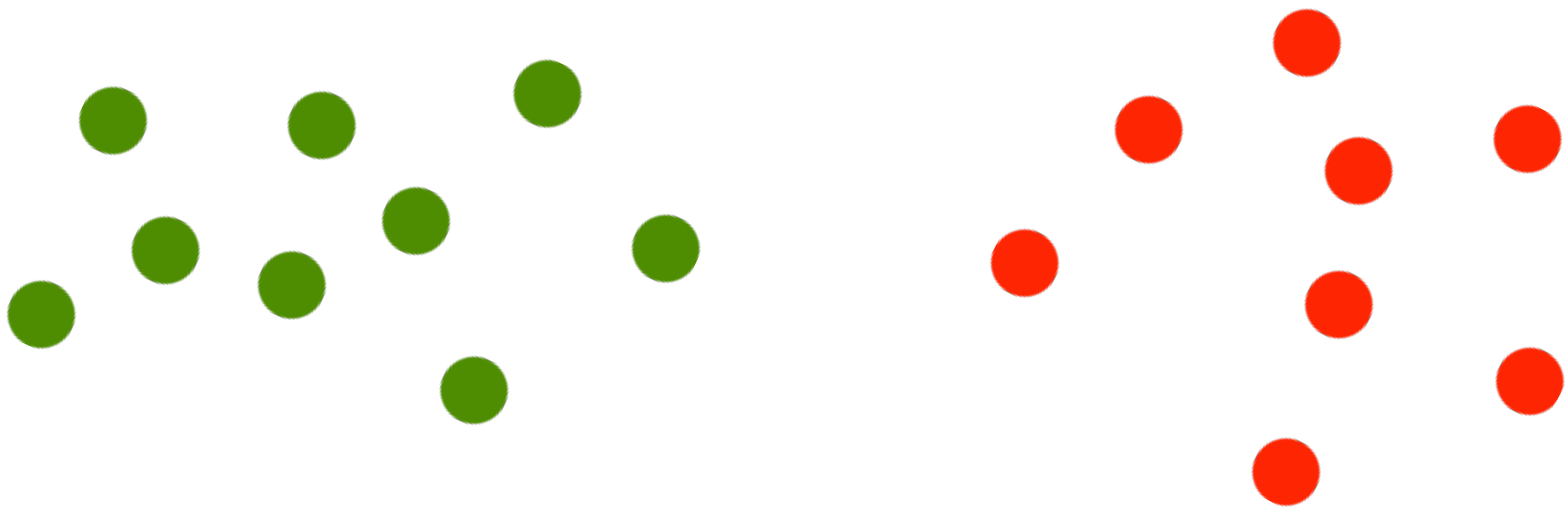


Linear separator: which one?

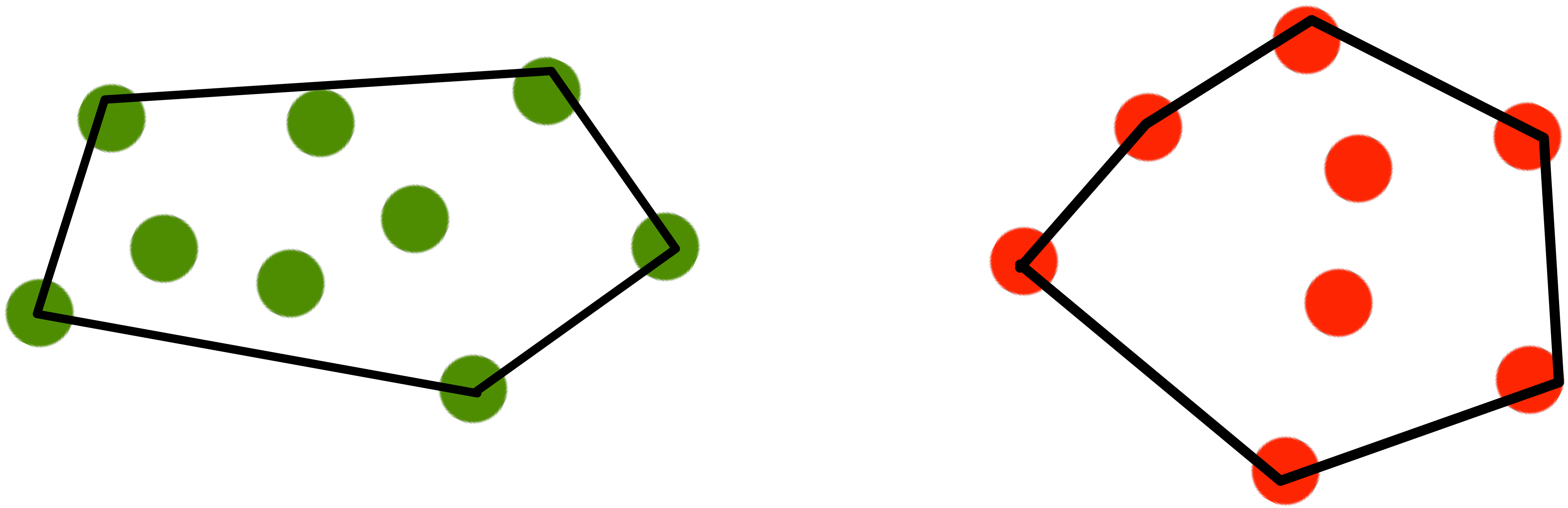


Just gimme some separator and get on with it…

Linear separators: geometric view



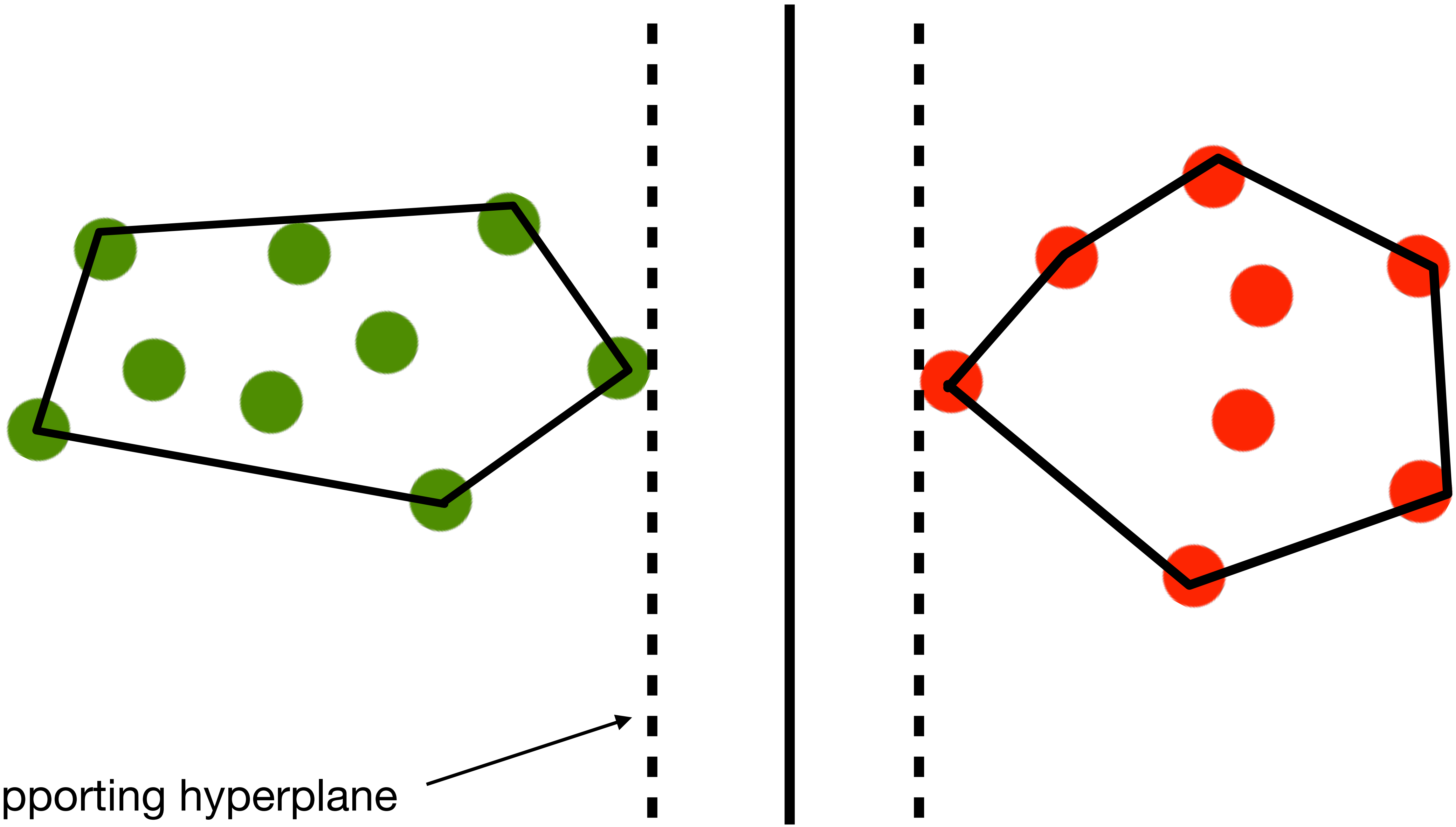
Linear separators: geometric view



Convex hulls of the respective points

https://en.wikipedia.org/wiki/Convex_hull

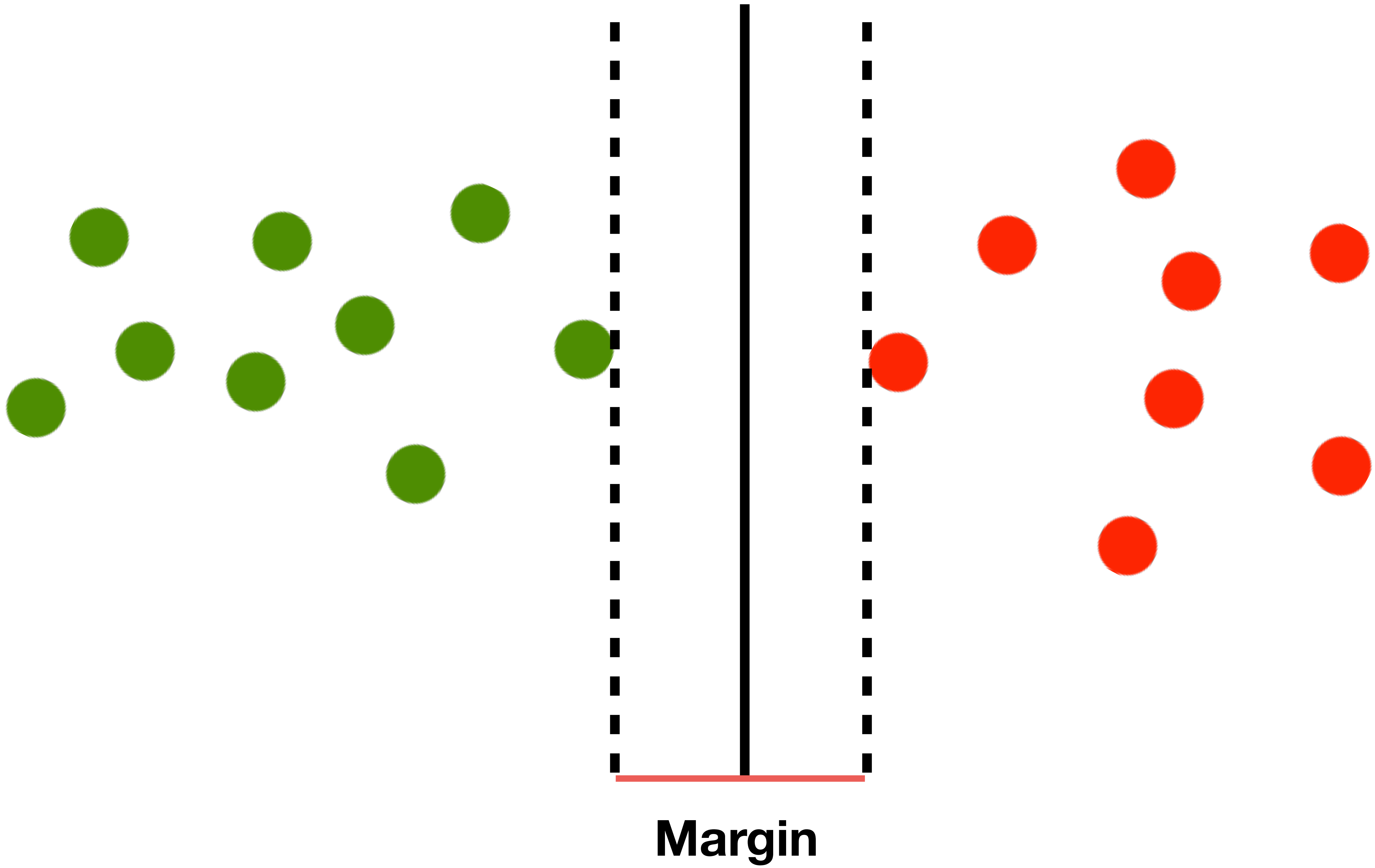
Linear separators: geometric view



Supporting hyperplane

https://en.wikipedia.org/wiki/Supporting_hyperplane

Linear separator with margin



The notion of margin

Suppose training data are strictly linearly separable

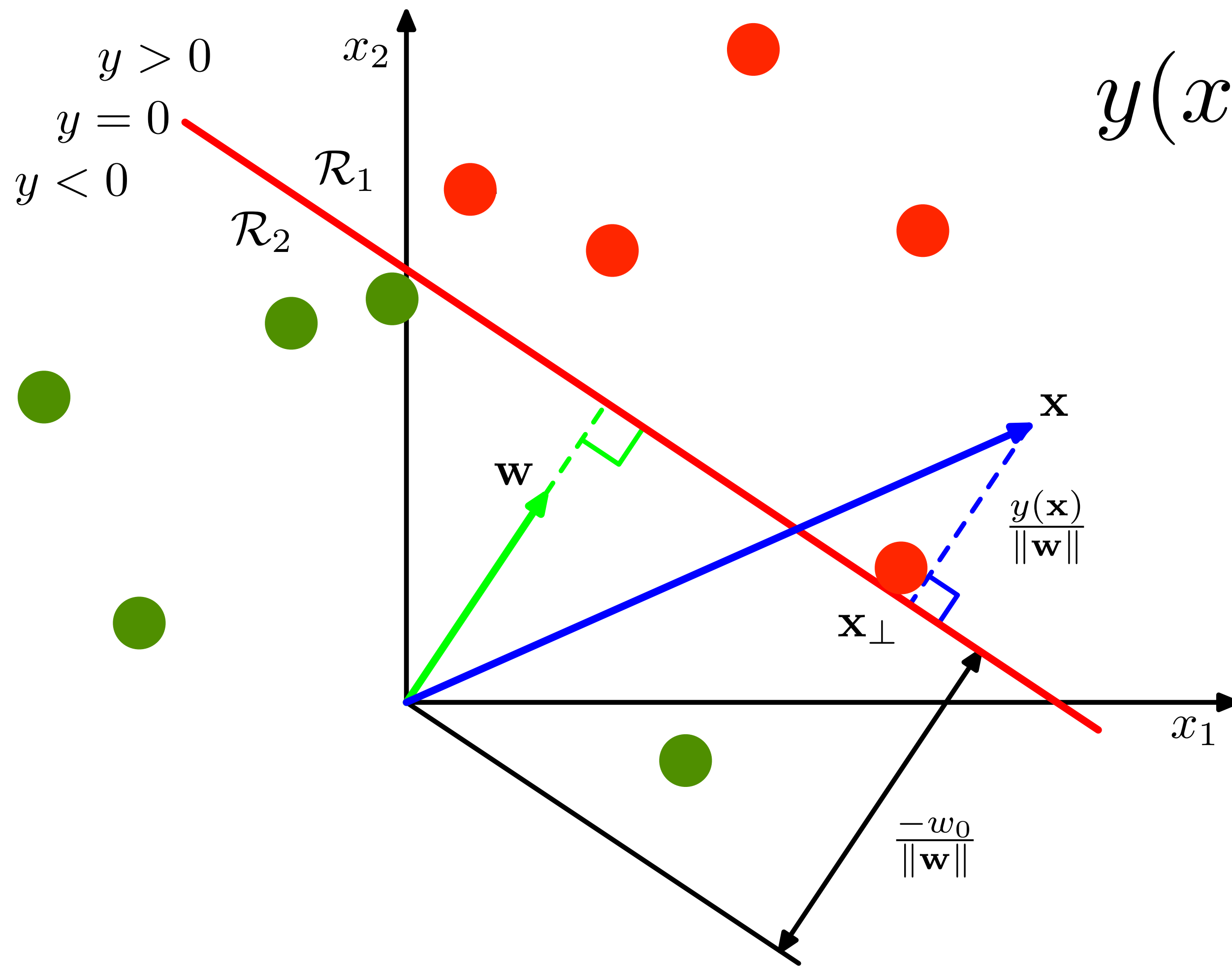
There exists (w, w_0) such that $w^T x + w_0 > 0$ for positive points and $w^T x + w_0 < 0$ for negative points *(assume data are 'bounded')*

Clearly, $(\delta w, \delta w_0)$ for any scalar $\delta > 0$ also works. So let us introduce **canonical hyperplane / normalization**

$$\min_{1 \leq i \leq N} |w^T x_i + w_0| = 1$$

Exercise: With this normalization, show that the point closest to the separating hyperplane is at a distance $1/\|w\|$

Recall: distance to hyperplane



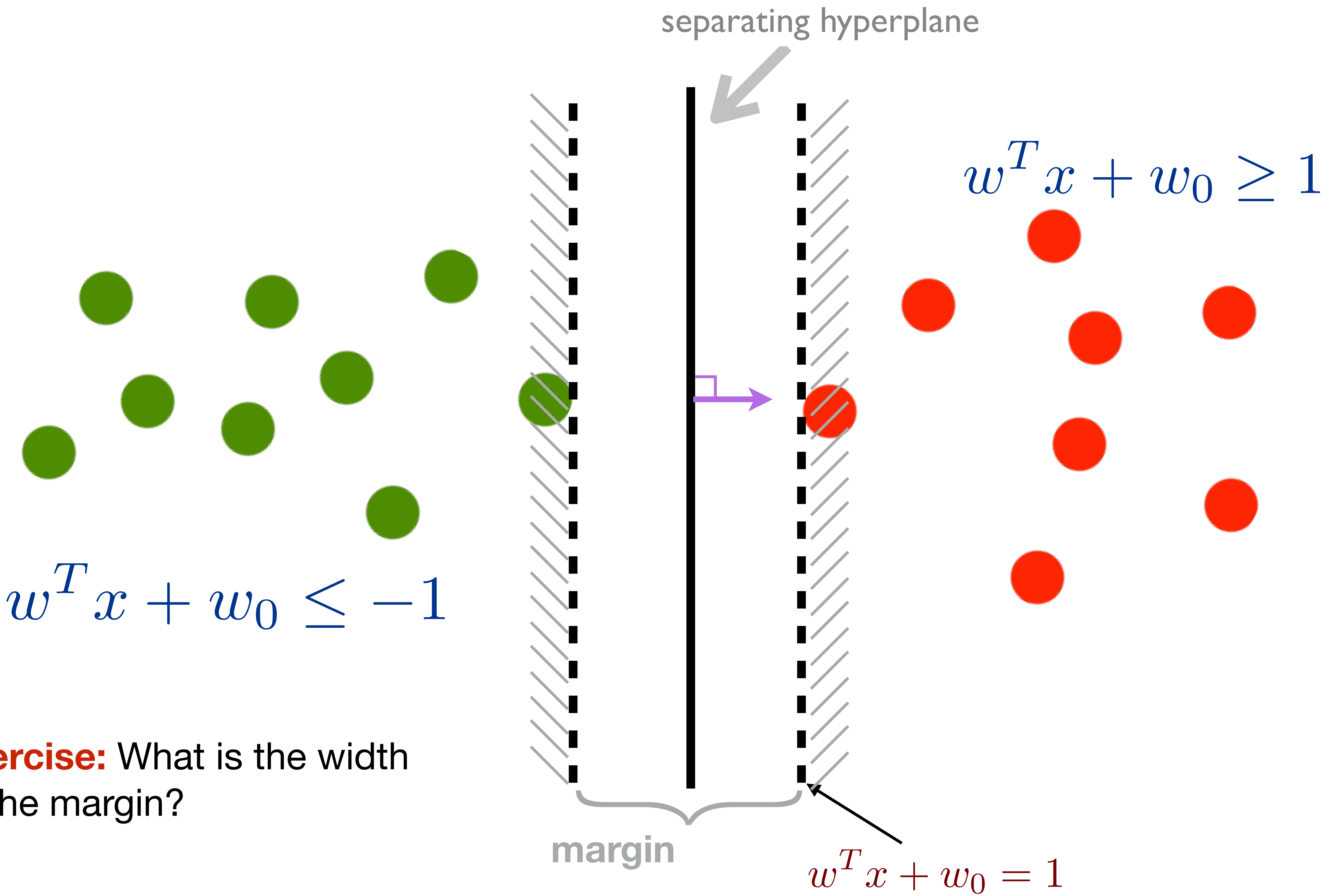
$$y(x) = w^T x + w_0$$

2D example
(Fig 4.1 in Bishop)

Recall: Write $x = x_\perp + \gamma \frac{w}{\|w\|}$ and conclude that γ is given by

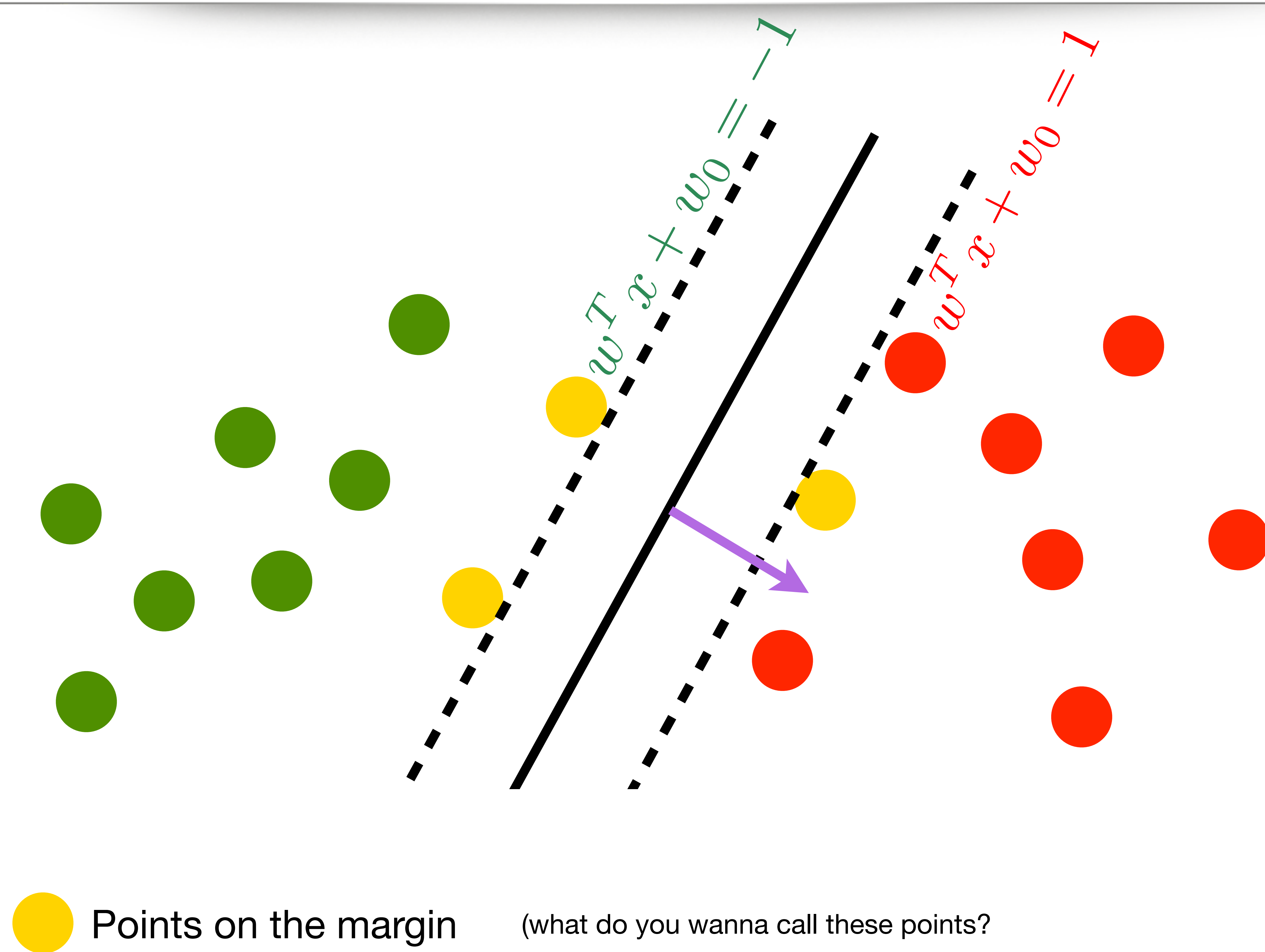
$$\gamma = \frac{w^T x + w_0}{\|w\|}$$ *(signed distance to the decision hyperplane)*

The notion of margin



Exercise: What is the width of the margin?

Margin based classifier

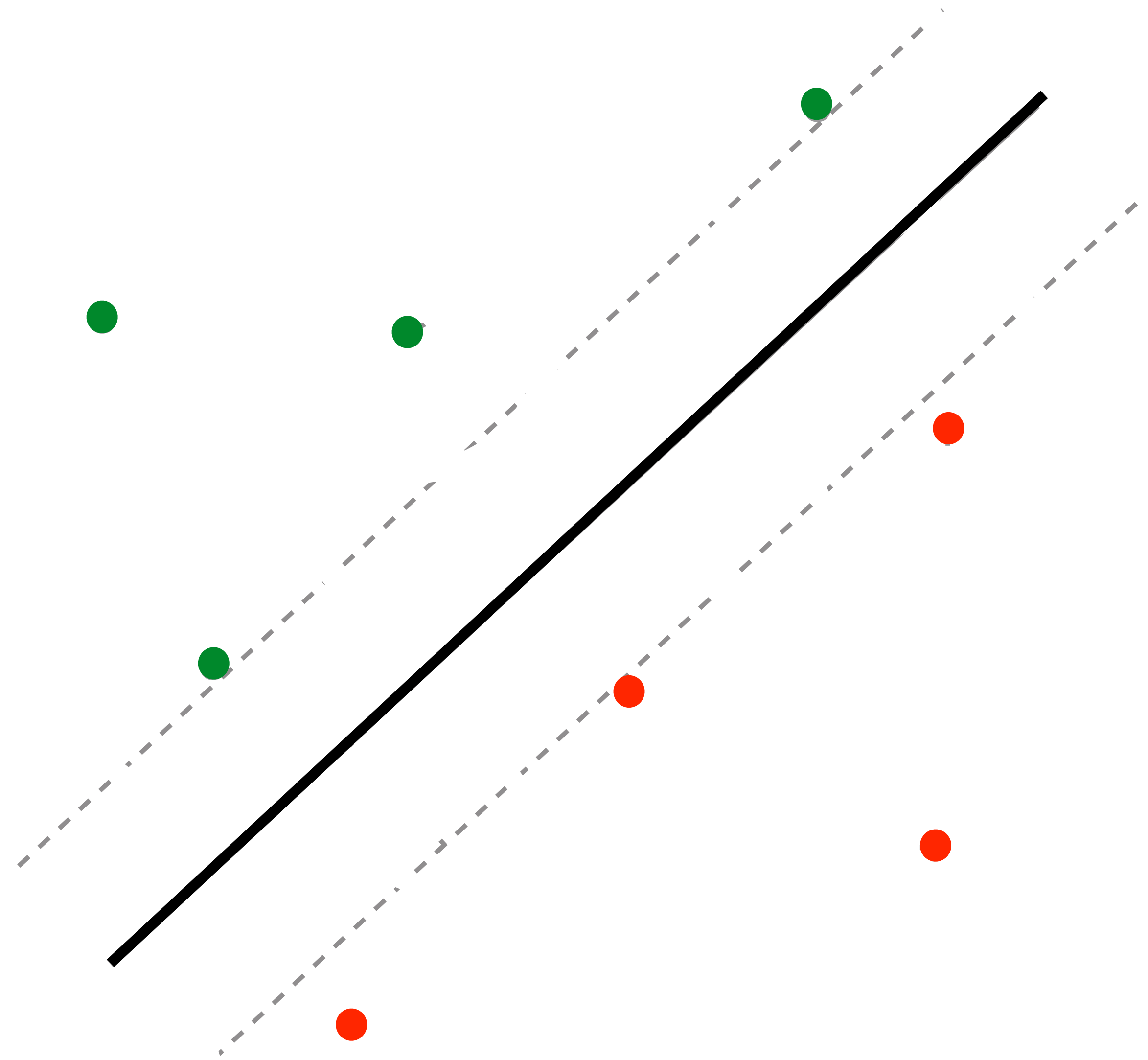


Points on the margin (what do you wanna call these points?

Why large margins?

Intuition: Suppose train and test points from same distribution

Except for some outliers, most test data points may lie close to training points



Why large margins?

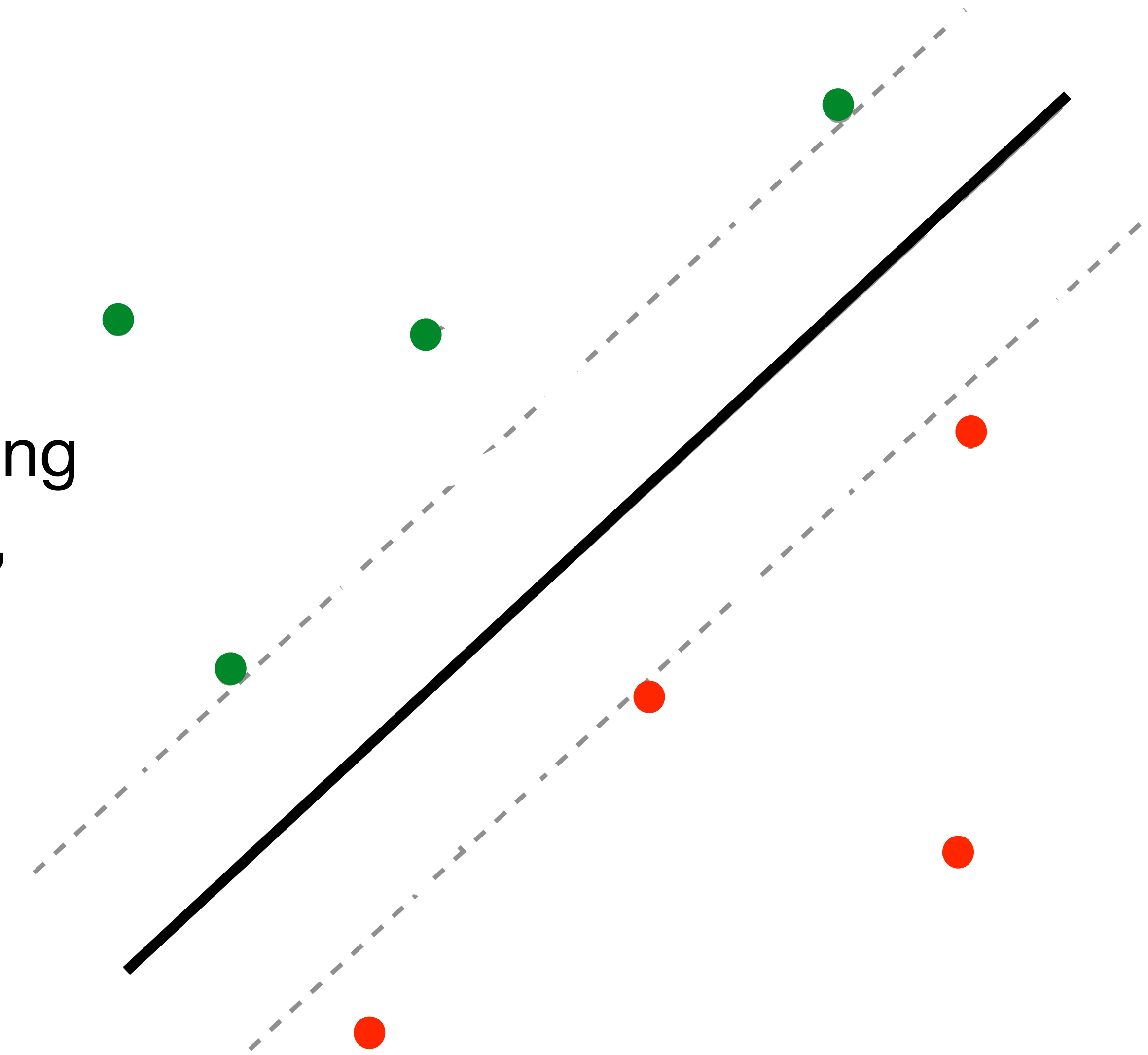
Intuition: Suppose train and test points from same distribution

Except for some outliers, most test data points may lie close to training points

Suppose test data generated by adding bounded noise to training data. Thus,

$$(x, y) \rightarrow (x + \delta x, y)$$

$$\|\delta x\| \leq r$$



Why large margins?

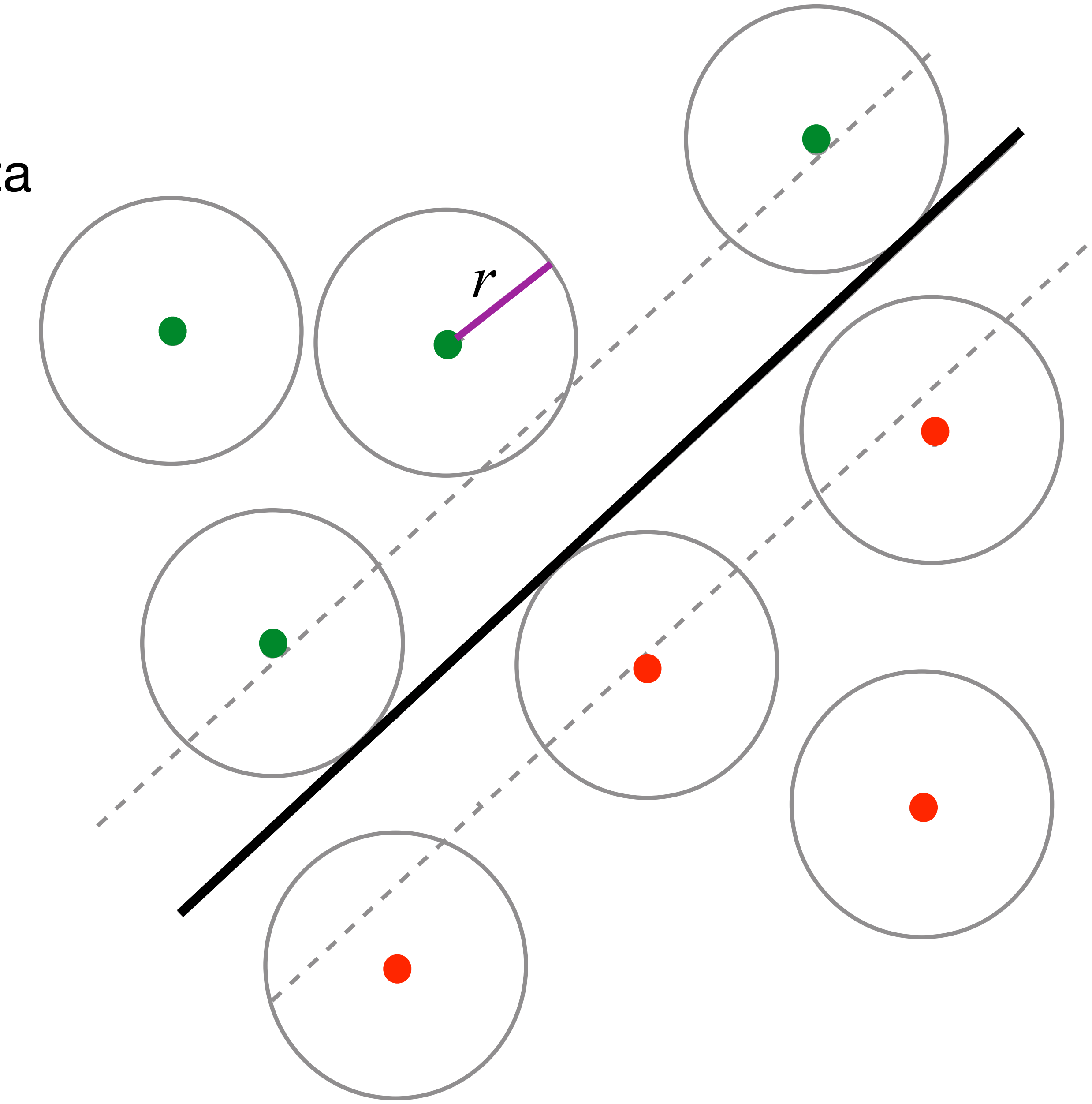
Intuition: Suppose train and test points from same distribution

Except for some outliers, most test data points may lie close to training points

Suppose test data generated by adding bounded noise to training data. Thus,

$$(x, y) \to (x + \delta x, y)$$

$$\|\delta x\| \leq r$$



Why large margins?

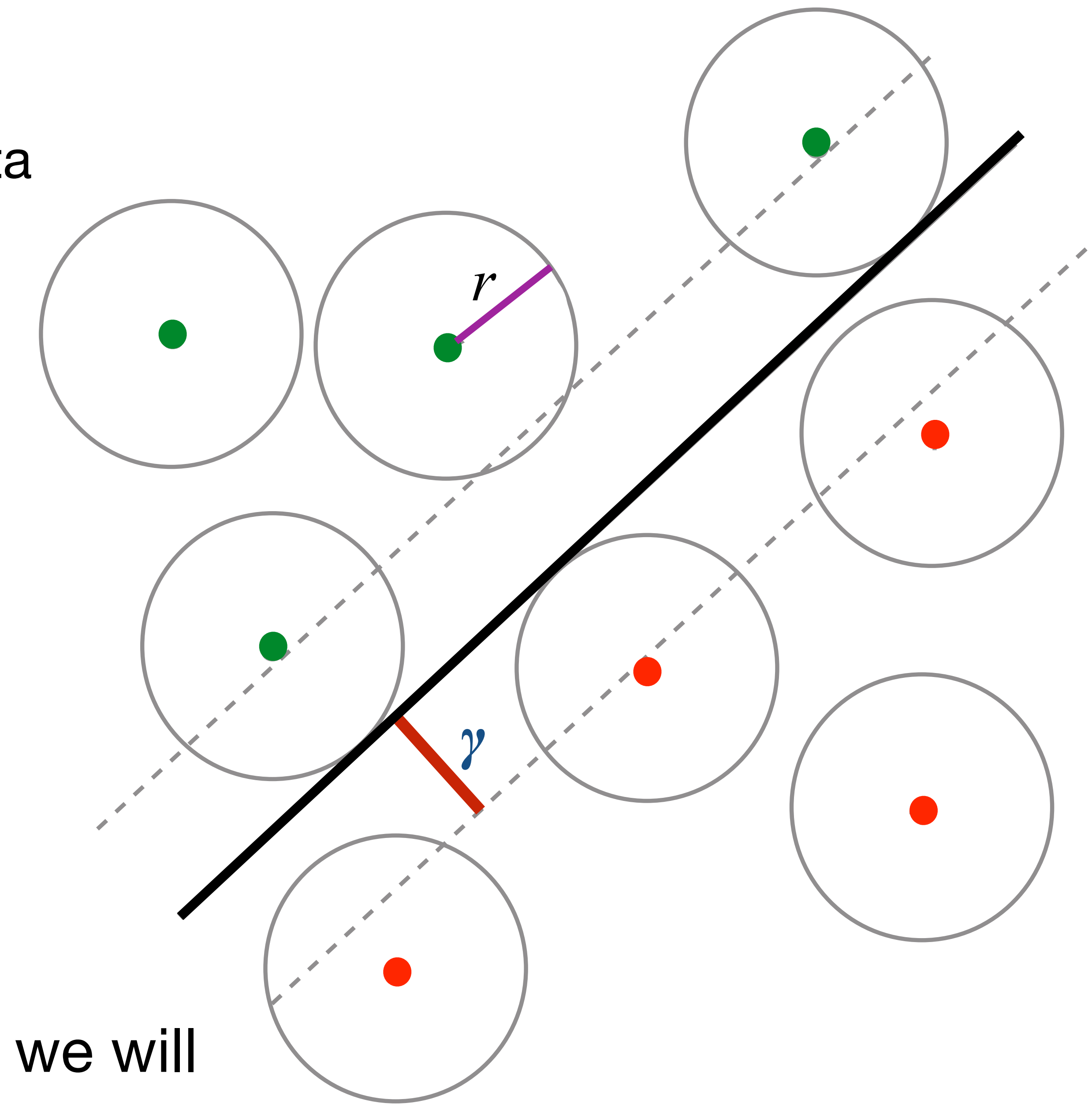
Intuition: Suppose train and test points from same distribution

Except for some outliers, most test data points may lie close to training points

Suppose test data generated by adding bounded noise to training data. Thus,

$$(x, y) \to (x + \delta x, y)$$

$$\|\delta x\| \leq r$$



If we manage to find a separating hyperplane with margin $\gamma > r$, then we will correctly classify **all** test data points

In other words, **robust** to any kind of noise that is bounded by *r* !

Why large margins?

This idea formalized in statistical learning theory.

We can prove a theorem of the form:

Prob(test point is misclassified) ≤ margin error + O(1 / margin)

Fraction of training data points
with margin smaller than $1/\|w\|$

Have we seen this idea before?

Why large-margins?

Prob[test point is misclassified] ≤ O(margin error) + O(1 / margin)

This is essentially a **bias-variance tradeoff**

Keep margin error small (overfitting) but that drives up the O(1/margin) term
Similarly, we can have a huge margin (underfitting) driving up margin error

A Unified Bias-Variance Decomposition and its Applications

Pedro Domingos PEDROD@CS.WASHINGTON.EDU
Department of Computer Science and Engineering, University of Washington, Seattle, WA 98195, U.S.A.

Abstract
This paper presents a unified bias-variance decomposition that is applicable to squared loss, zero-one loss, variable misclassification costs, and other loss functions. The unified predictions fluctuate in response to the training set. Tibshirani (1996) defines bias and variance, but decomposes loss into bias and the "aggregation effect," a quantity unrelated to his definition of variance. James and Hastie (1997) extend this approach by defining

(circa 2000)

Bias-Variance Analysis of Support Vector Machines for the Development of SVM-Based Ensemble Methods

Giorgio Valentini VALENTINI@DSI.UNIMI.IT
DSI - Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
Via Comelico 39, Milano, Italy

Thomas G. Dietterich TGD@CS.ORST.EDU
Department of Computer Science
Oregon State University
Corvallis, OR 97331, USA

(circa 2004)

Finding a large-margin hyperplane

Linearly separable case

We want a large margin, i.e., maximize 1/||w||

Canonical hyperplane $$\min_{1 \leq i \leq N} |w^T x_i + w_0| = 1$$

Thus, for all the training data points we will have

$$y_i(w^T x_i + w_0) \geqslant 1, \quad 1 \leqslant i \leqslant N.$$

Naive formulation:

$$\max_{w, w_0} \frac{1}{\|w\|}$$

$$\min_{1 \leq i \leq N} y_i(w^T x_i + w_0) = 1.$$

SVM: linearly separable data

Slightly better formulation

$$\cancel{\max_{w,w_0} \quad \frac{1}{\|w\|}}$$

$$\cancel{y_i(w^T x_i + w_0) \geqslant \ 1, \quad 1 \leqslant i \leqslant N}$$

Convex formulation

$$\min_{w,w_0} \quad \frac{1}{2}\|w\|^2$$

$$y_i(w^T x_i + w_0) \geqslant \ 1, \quad 1 \leqslant i \leqslant N$$

Hard-margin SVM: solution

Lagrangians and KKT conditions

$$L(w, w_0, \alpha) := \tfrac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(w^T x_i + w_0) - 1].$$

KKT conditions

(stationarity) $$\frac{\partial L}{\partial w} = 0, \ \frac{\partial L}{\partial w_0} = 0$$

(complementarity) $$\alpha_i [y_i(w^T x_i + w_0) - 1] = 0, \ \forall i.$$

(primal feasibility) $$y_i(w^T x_i + w_0) \geq 1, \ \forall i$$

(dual feasibility) $$\alpha_i \geq 0, \ \forall i$$

These conditions reveal a lot about the SVM problem

[See Chapter 5 of Boyd and Vandenberghe's Convex Optimization if you are rusty / unfamiliar with KKT conditions and constrained optimization]

Hard-margin SVM: solution

Lagrangians and KKT conditions

$$L(w, w_0, \alpha) := \tfrac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(w^T x_i + w_0) - 1].$$

Stationarity $\frac{\partial L}{\partial w} = 0, \ \frac{\partial L}{\partial w_0} = 0$ implies

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0.$$

The optimal hyperplane is a linear combination of the training data!

(This is a very cool property; we'll see it again somepoint)

Hard-margin SVM: solution

What does complementarity imply? *(Hint: see picture)*

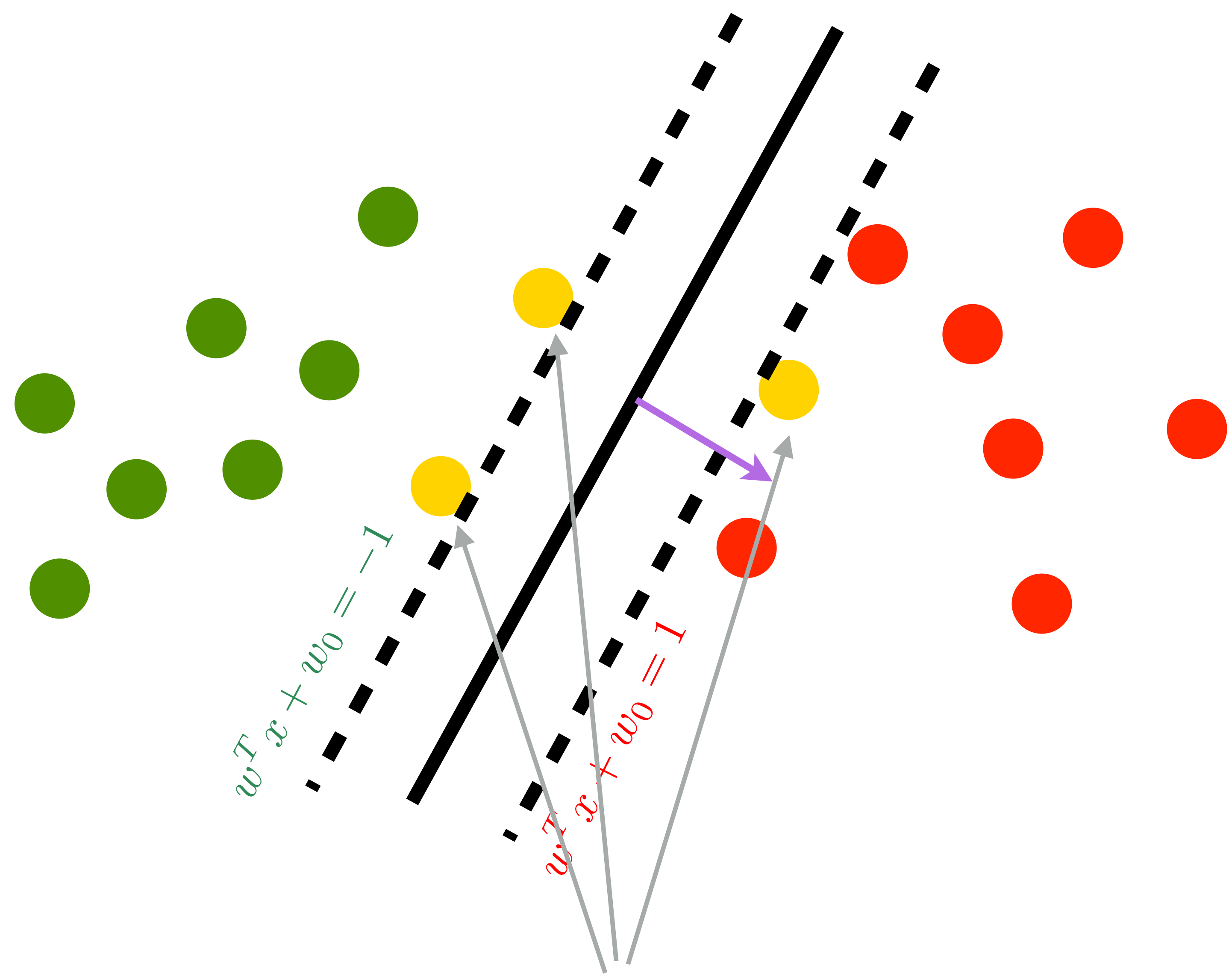
$$\alpha_i[y_i(w^T x_i + w_0) - 1] = 0, \;\; \forall i.$$

$$\alpha_i \geq 0, \;\; \forall i$$

If $\alpha_i > 0$, then point must lie on the margin (i.e., constraint is tight)

Thus, for the optimal hyperplane:

$$w = \sum_{i:\alpha_i > 0} \alpha_i y_i x_i.$$

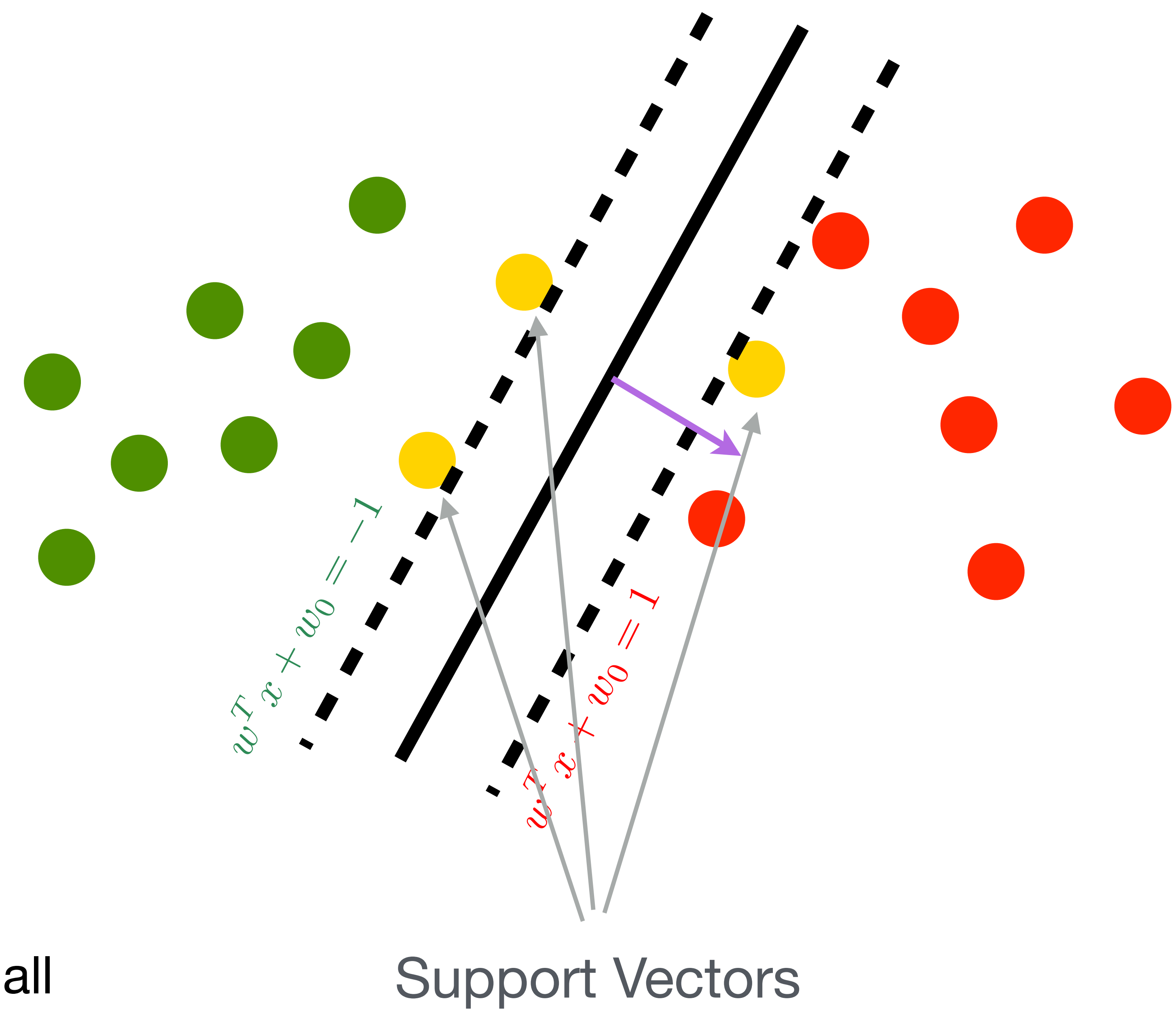


Hard-margin SVM: solution

If $\alpha_i > 0$, then point must lie on the margin (i.e., constraint is tight)

Thus, for the optimal hyperplane:

$$w = \sum_{i:\alpha_i>0} \alpha_i y_i x_i.$$



Exercise: Argue that if we were to drop all other points, the optimal hyperplane would still be the same!

But we are still missing something!

How to find the support vectors?
What if the data are not separable?

Finding support vectors

$$L(w, w_0, \alpha) := \tfrac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(w^T x_i + w_0) - 1].$$

SVM dual

$$\begin{aligned} \max_{\alpha \geq 0} \ & \left[g(\alpha) := \min_{w, w_0} \ L(w, w_0, \alpha) \right] \\ & = -\tfrac{1}{2} \left\| \sum\nolimits_i \alpha_i y_i x_i \right\|^2 + \sum\nolimits_i \alpha_i \\ & \sum\nolimits_i y_i \alpha_i = 0. \end{aligned}$$

Solve using `sklearn`, LIBSVM, etc.

Note: SVM optimization has great historical significance: it brought the field of nonlinear optimization to high prominence inside machine learning.

Finding support vectors

What about our good old loss-function viewpoint?

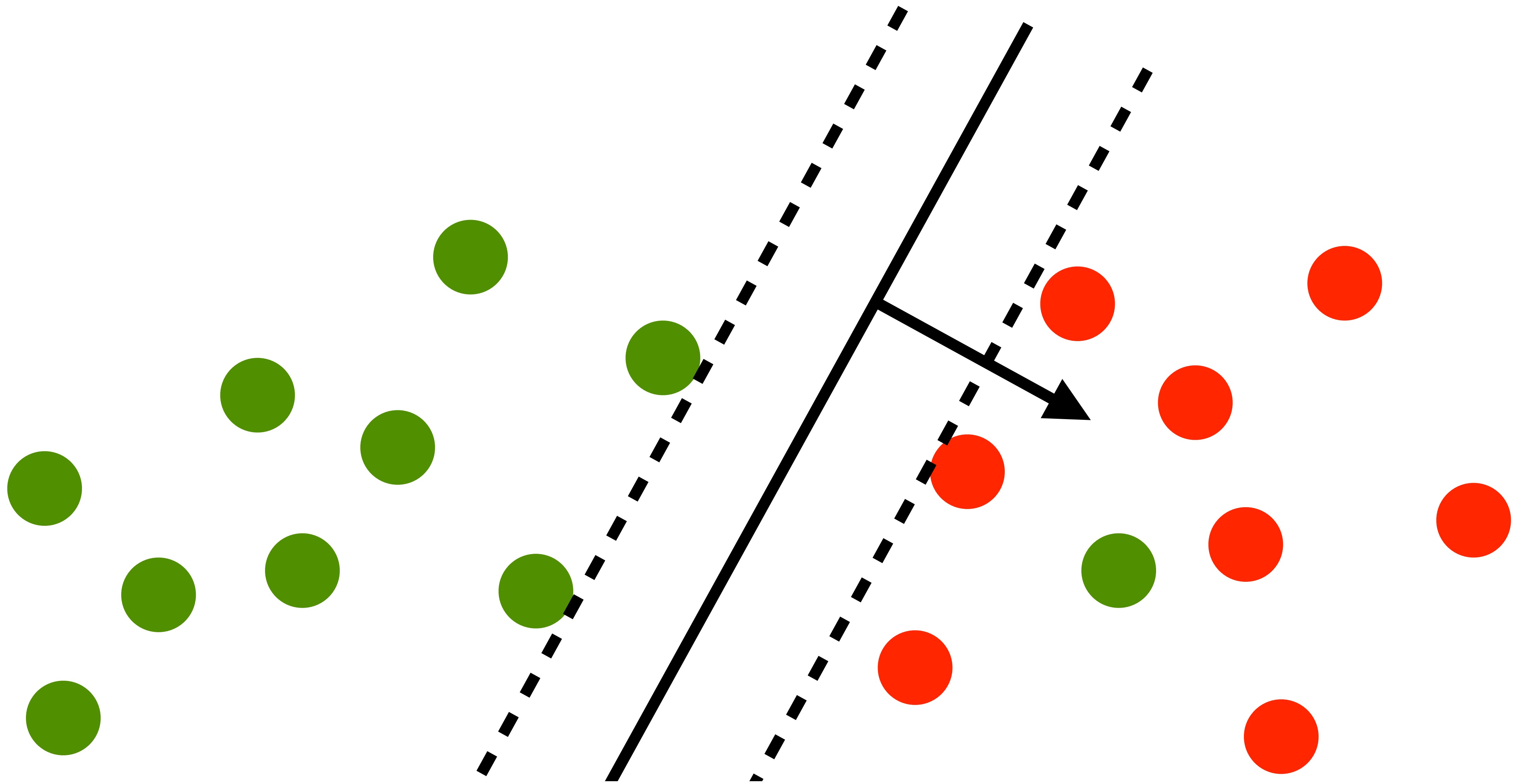
Exercise: Write SVM as an **Empirical Risk Minimization** problem

$$R_{\text{emp}}(w, w_0) := \sum_{i=1}^{N} \ell(y_i(w^T x_i + w_0))$$

Hint: Hinge loss is thy friend; if you don't see it, stay tuned for ideas!

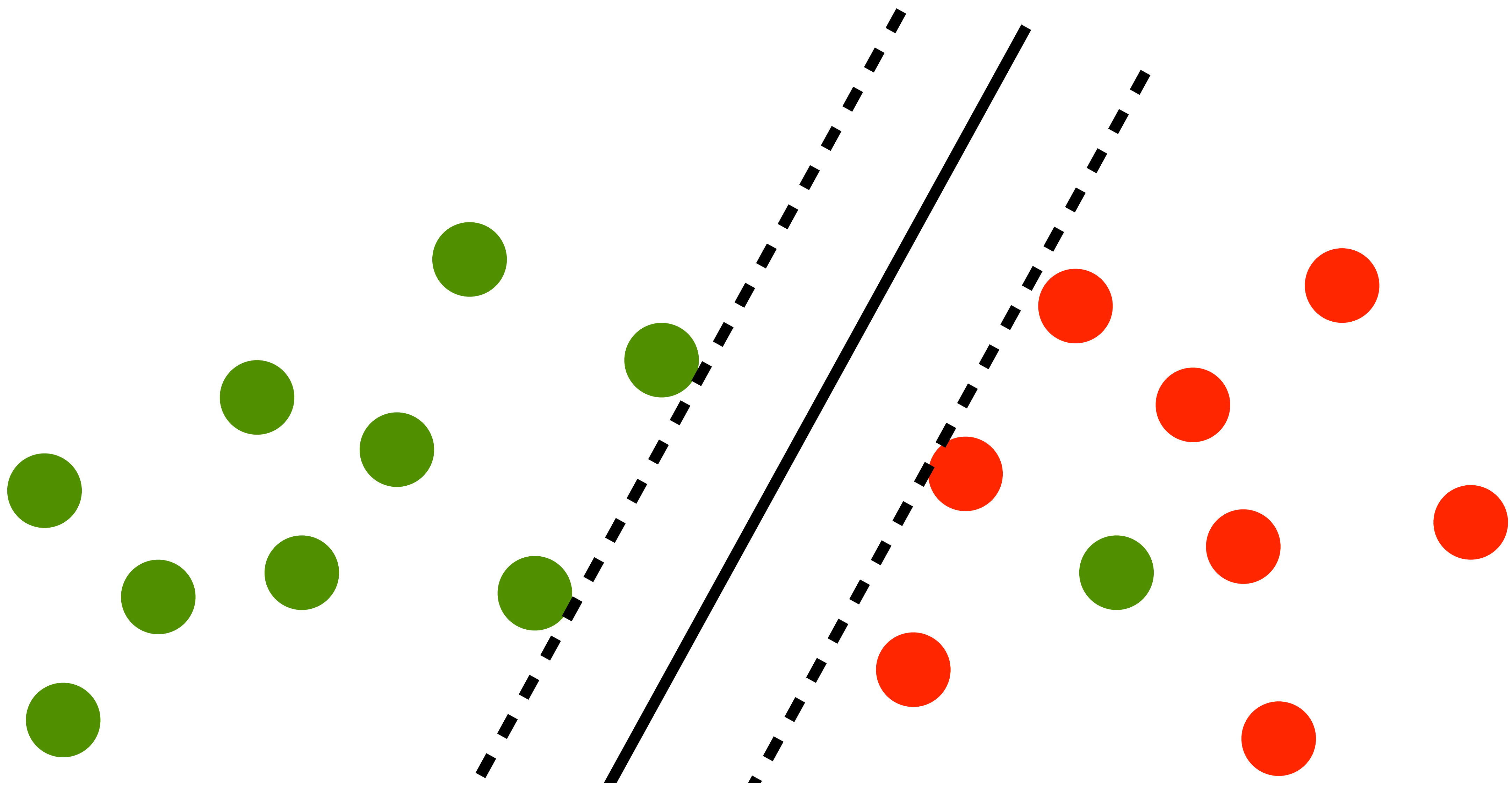
There was a time, when this exercise was worth several research papers!

Linearly inseparable data



linear separator is impossible

Linearly inseparable data



minimum error separator
is impossible (ok, it's just NP-Hard!)

Linear inseparable case

Convex formulation

$$\min_{w,w_0} \quad \frac{1}{2}\|w\|^2$$

$$y_i(w^T x_i + w_0) \geqslant 1, \quad 1 \leqslant i \leqslant N$$

Cannot satisfy all these requirements

Loosen the hard constraints by adding slacks

$$\min_{w,w_0,\xi} \quad \tfrac{1}{2}\|w\|^2 + C\sum\nolimits_i \xi_i$$

$$y_i(w^T x_i + w_0) \geqslant 1 - \xi_i, \quad 1 \leqslant i \leqslant N$$

$$\xi_i \geq 0, \quad 1 \leqslant i \leqslant N$$

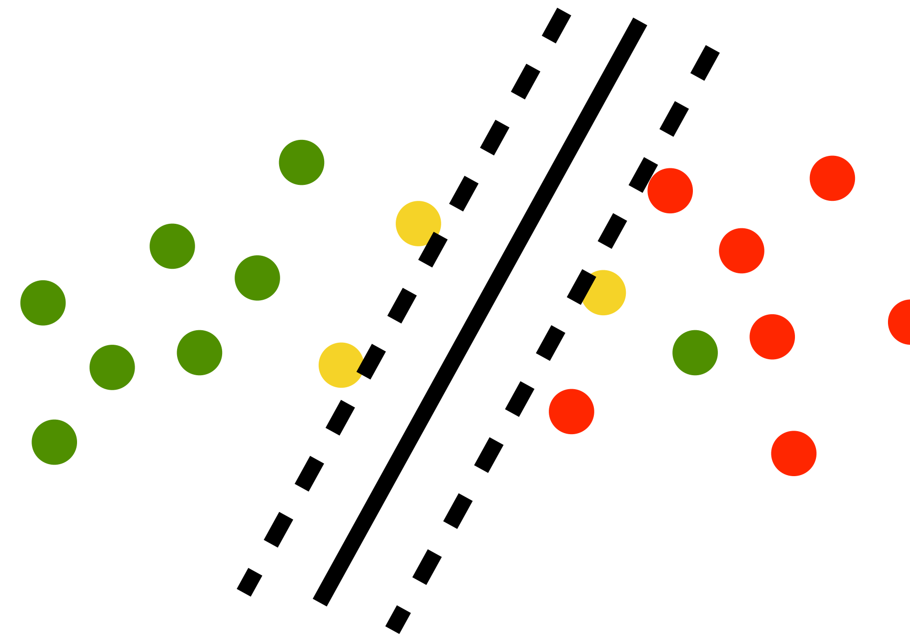
C: Hyperparameter, tells how soft (small C) or hard (large C)

Linear inseparable case

$$\min_{w,w_0,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$y_i(w^T x_i + w_0) \geqslant 1 - \xi_i, \quad 1 \leqslant i \leqslant N$$

$$\xi_i \geq 0, \quad 1 \leqslant i \leqslant N$$



Observations

- ✓ Whenever $\xi=0$, margin constraint is met (so x_i is not a margin error)
- ✓ All nonzero ξ correspond to margin errors / violations
- ✓ Above formulation makes tradeoff between margin width and margin errors apparent
- ✓ Amount by which we decrease or increase importance of training errors controlled by C

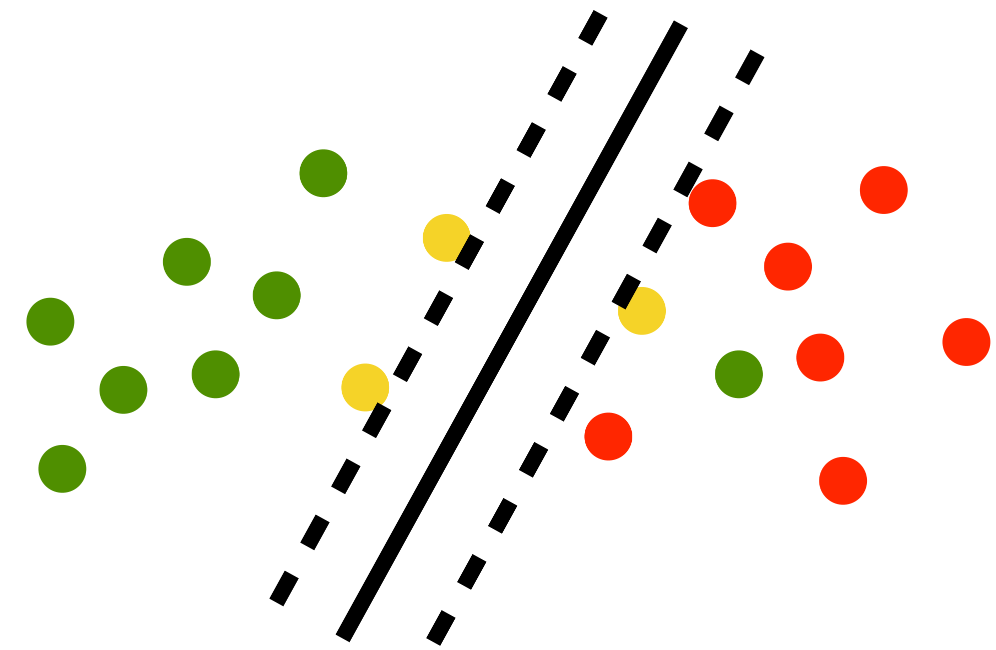
Dual of soft-SVM

After some simplification it can be shown that the dual of the soft SVM is:

$$\max_{\alpha} \quad -\tfrac{1}{2}\left\|\sum\nolimits_i \alpha_i y_i x_i\right\|^2 + \sum\nolimits_i \alpha_i$$

$$\sum\nolimits_i y_i \alpha_i = 0$$

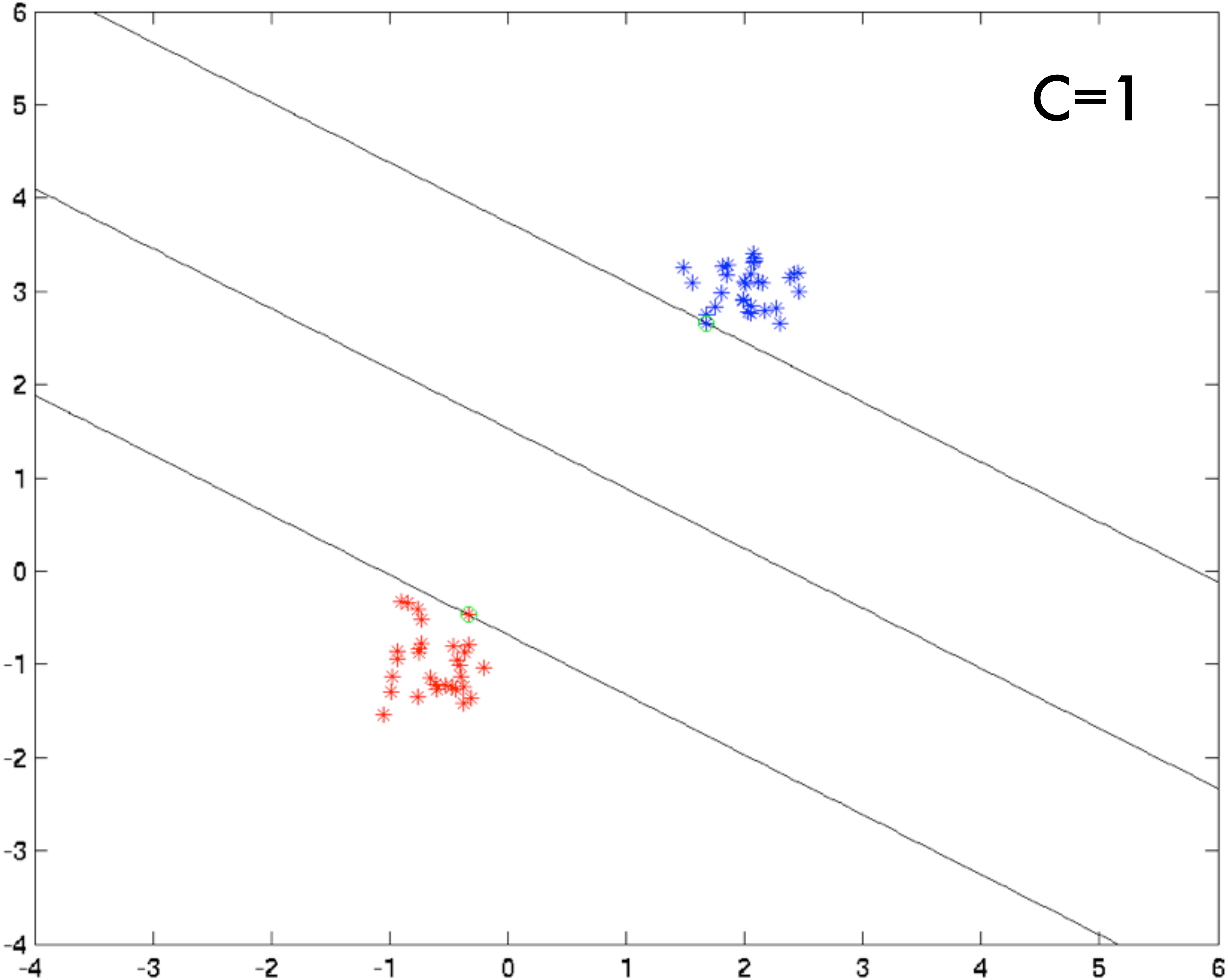
$$0 \leq \alpha \leq C.$$

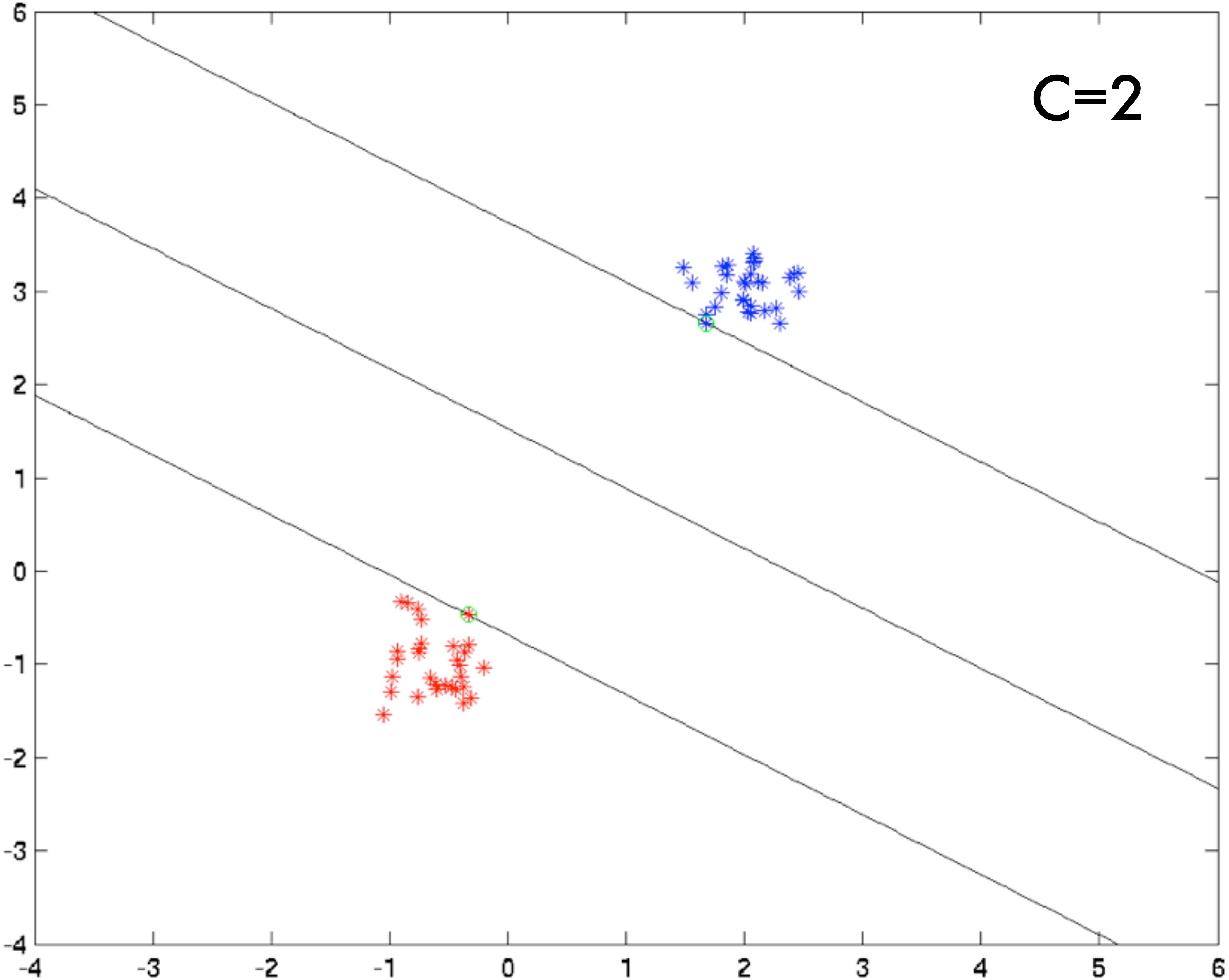


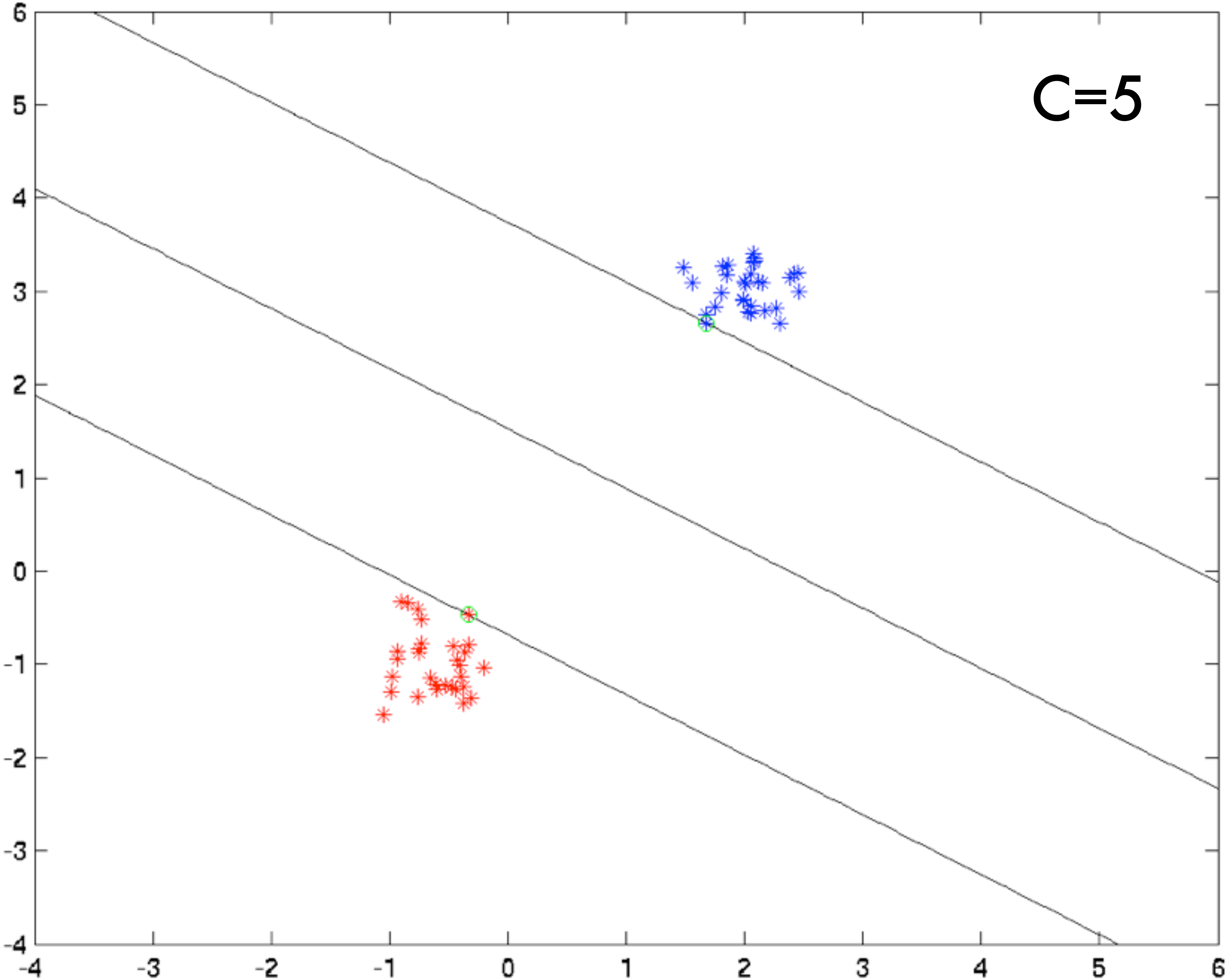
Exercise: Conclude via the complementarity conditions that

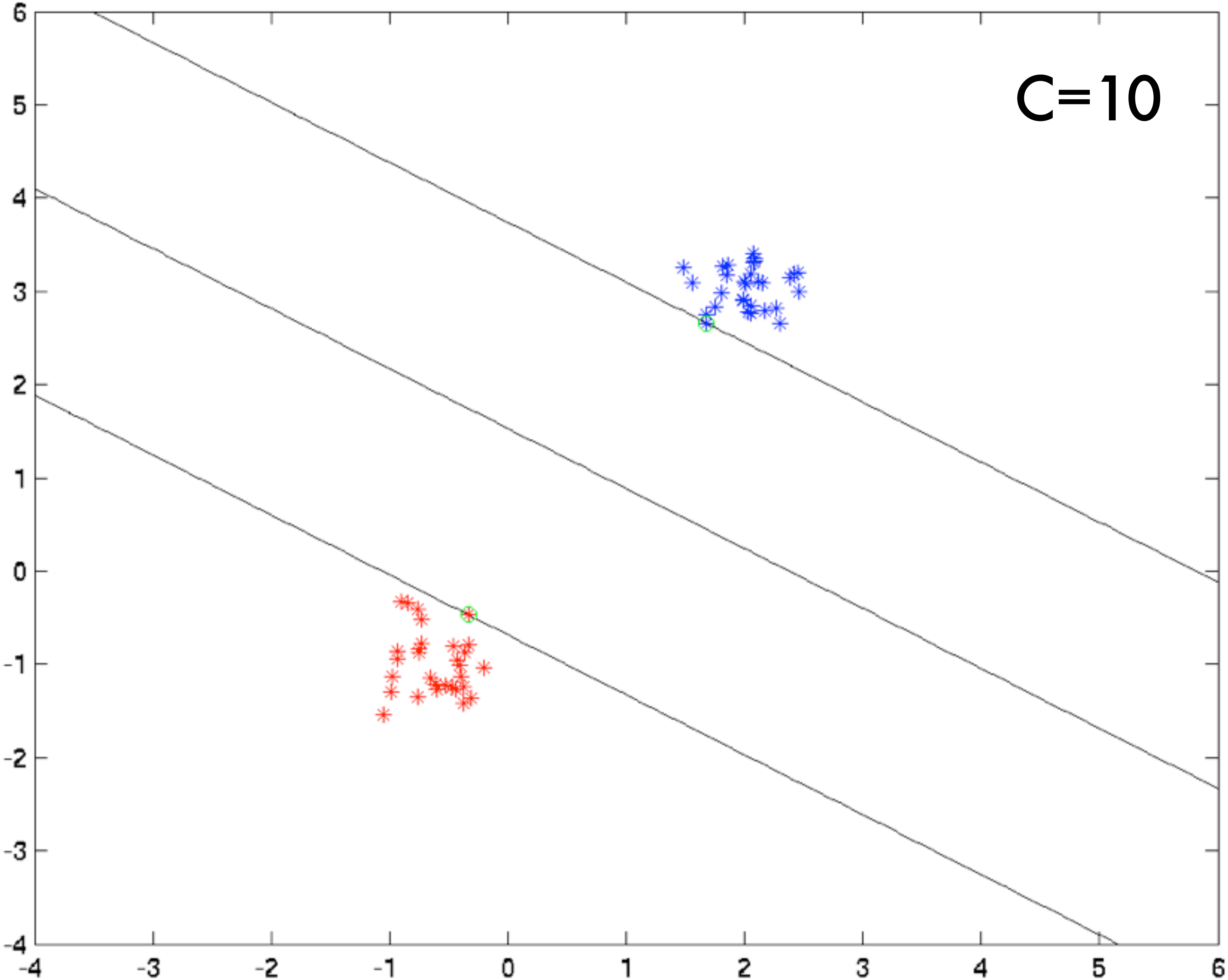
$$\begin{aligned}
\alpha_i = 0 &\implies y_i(w^T x_i + w_0) \geqslant 1 && \text{(correctly classified)}\\
\alpha_i = C &\implies y_i(w^T x_i + w_0) \leqslant 1 && \text{(margin violation)}\\
0 < \alpha_i < C &\implies y_i(w^T x_i + w_0) = 1 && \text{(support vector)}
\end{aligned}$$

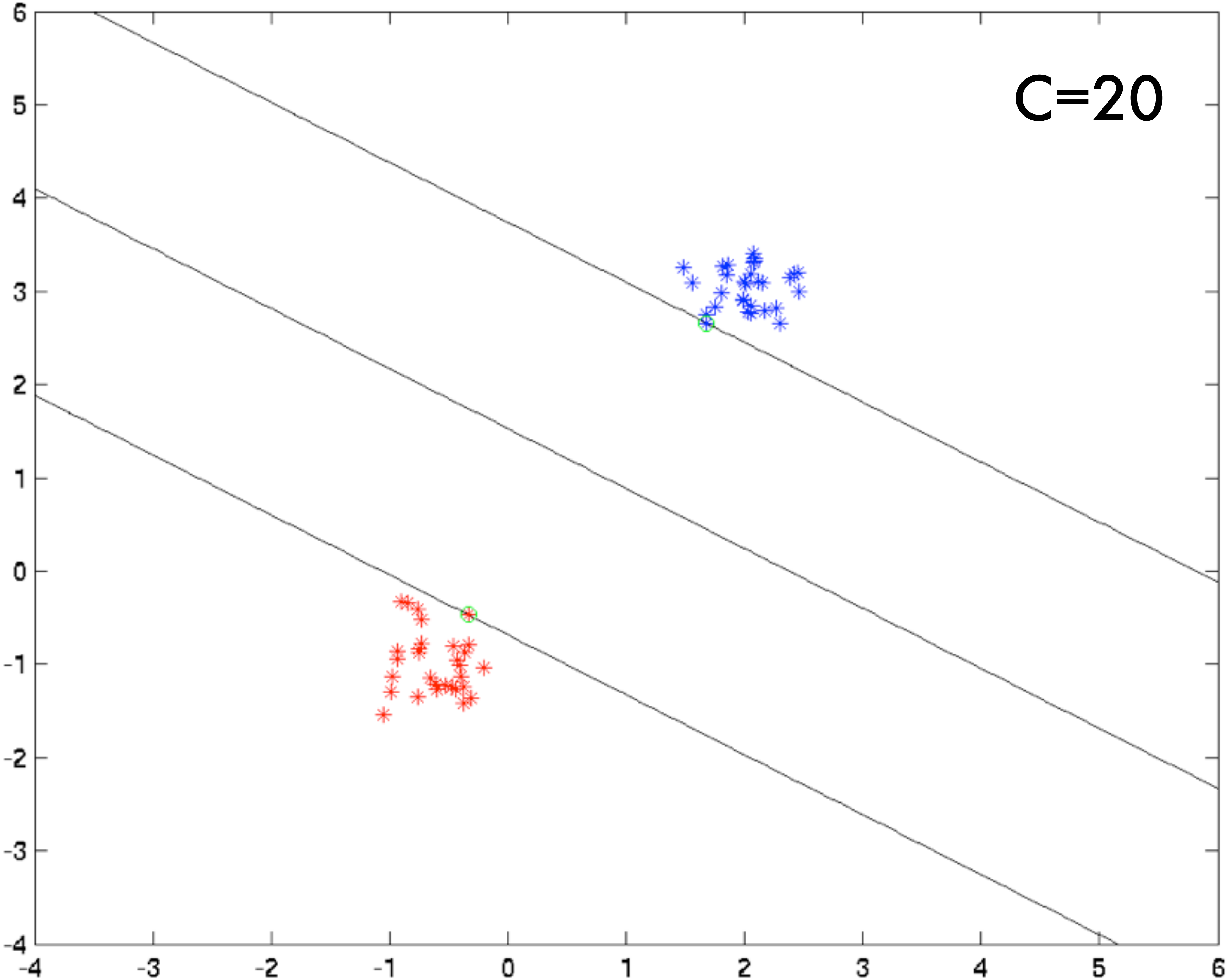
[compare with similar analysis for hard-SVM]

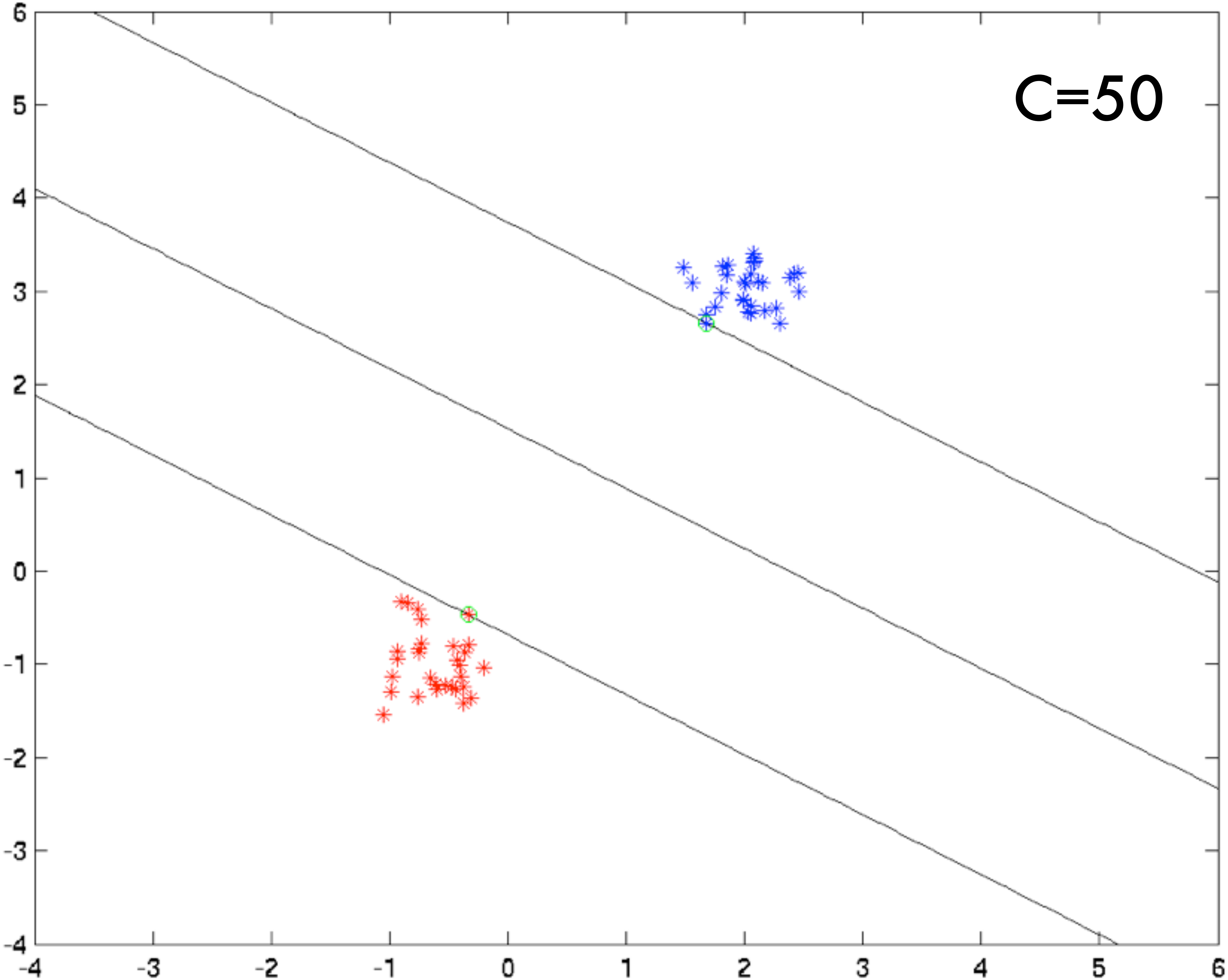

C=1
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6


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

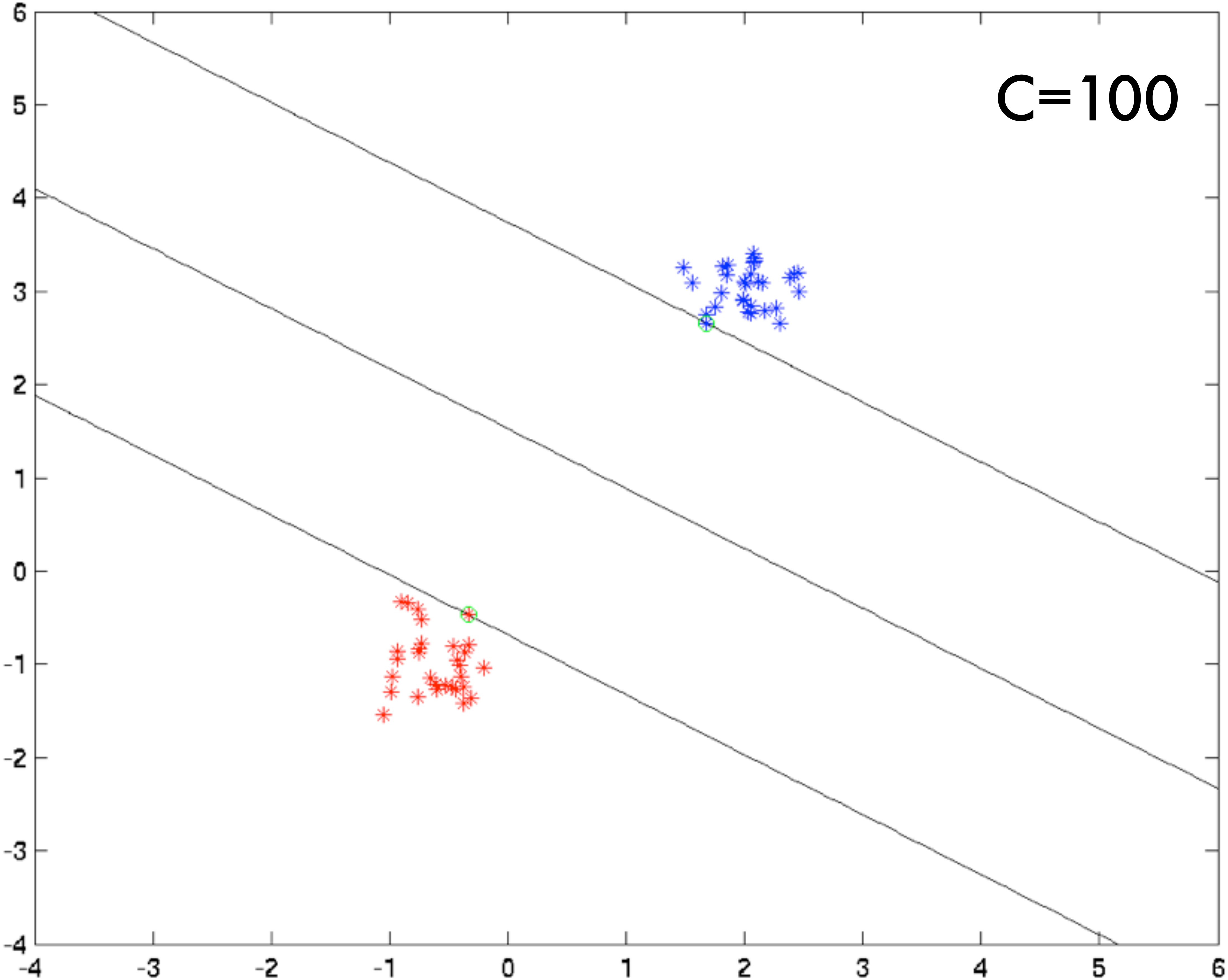

C=5
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

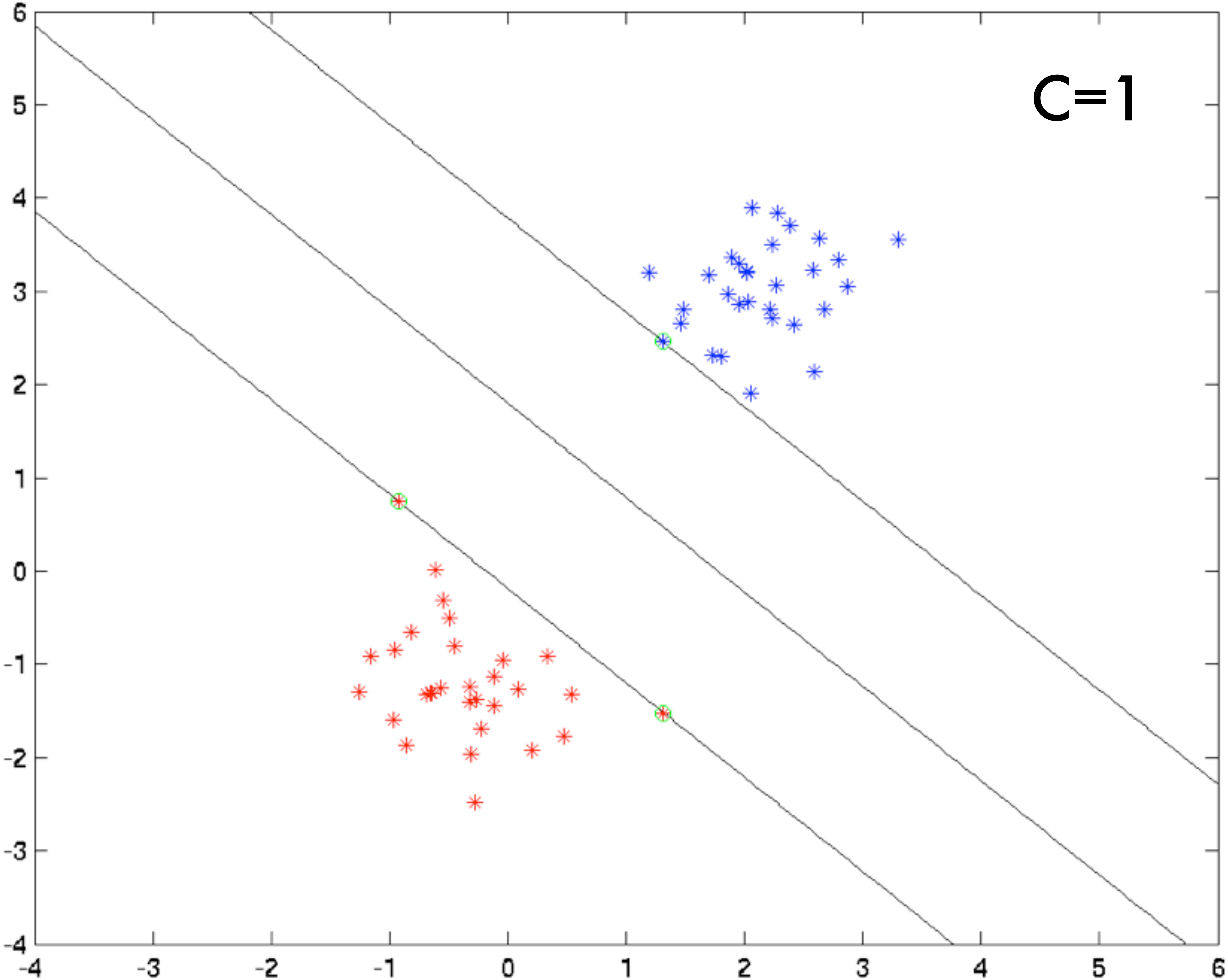

C=10
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

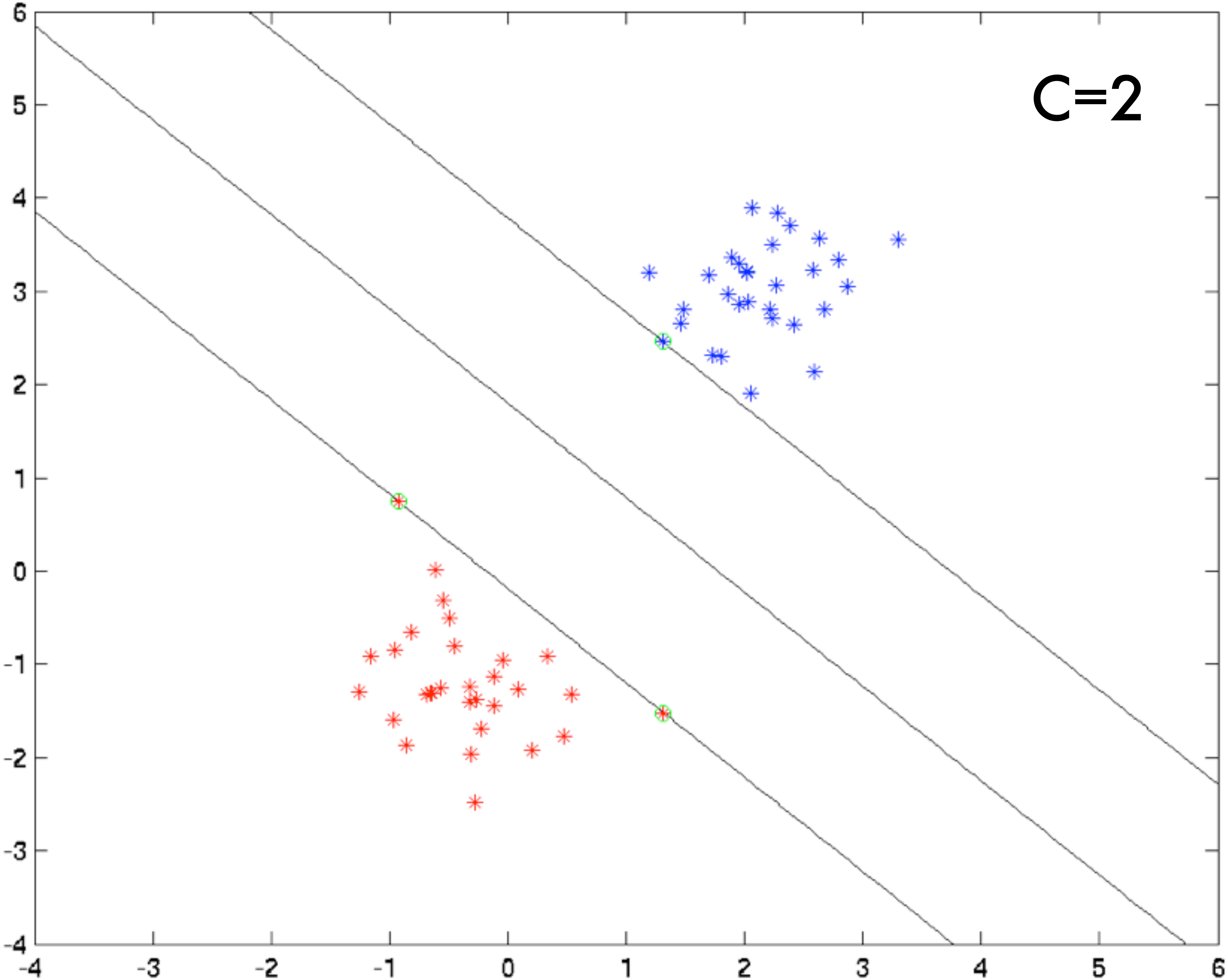

C=20
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

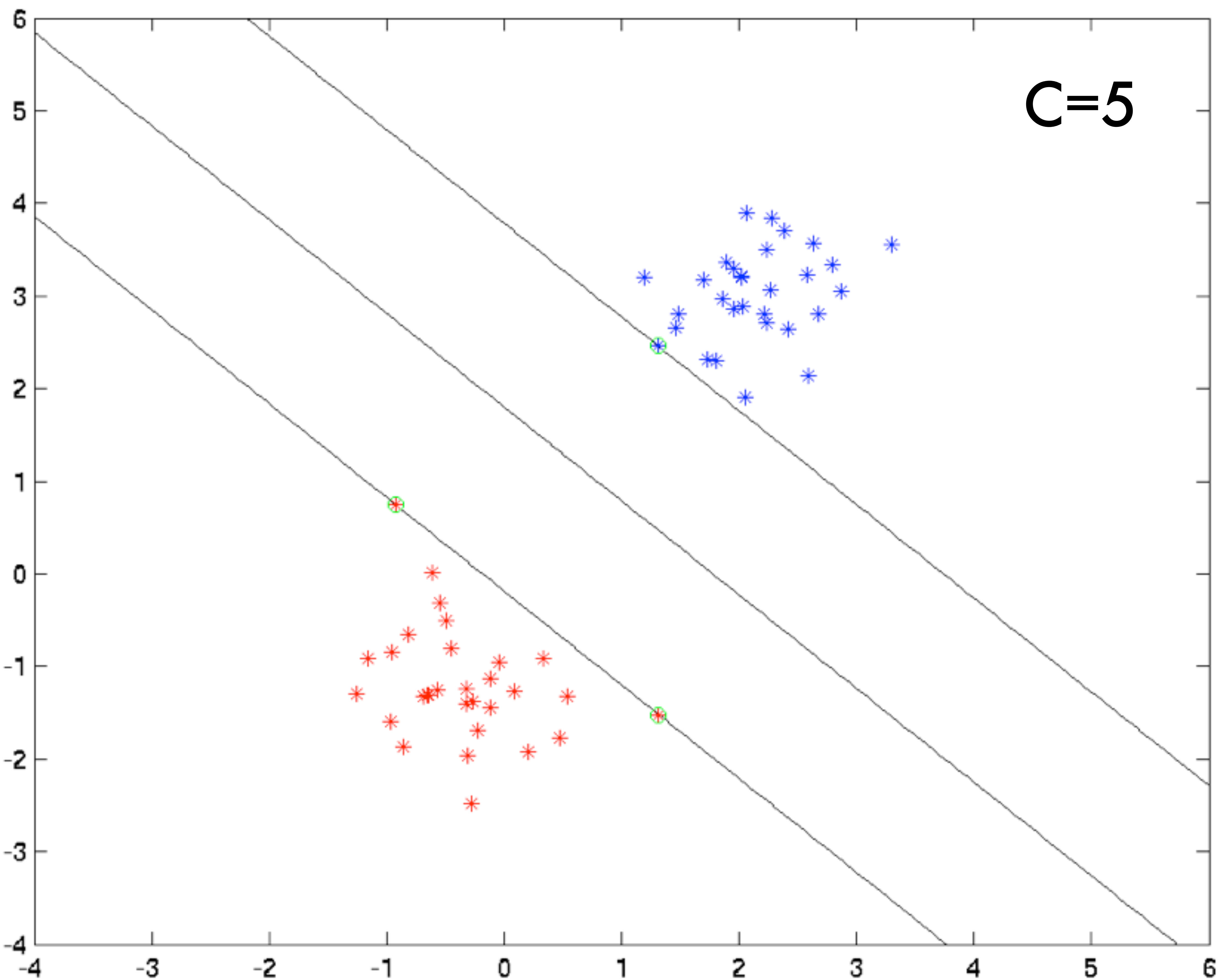


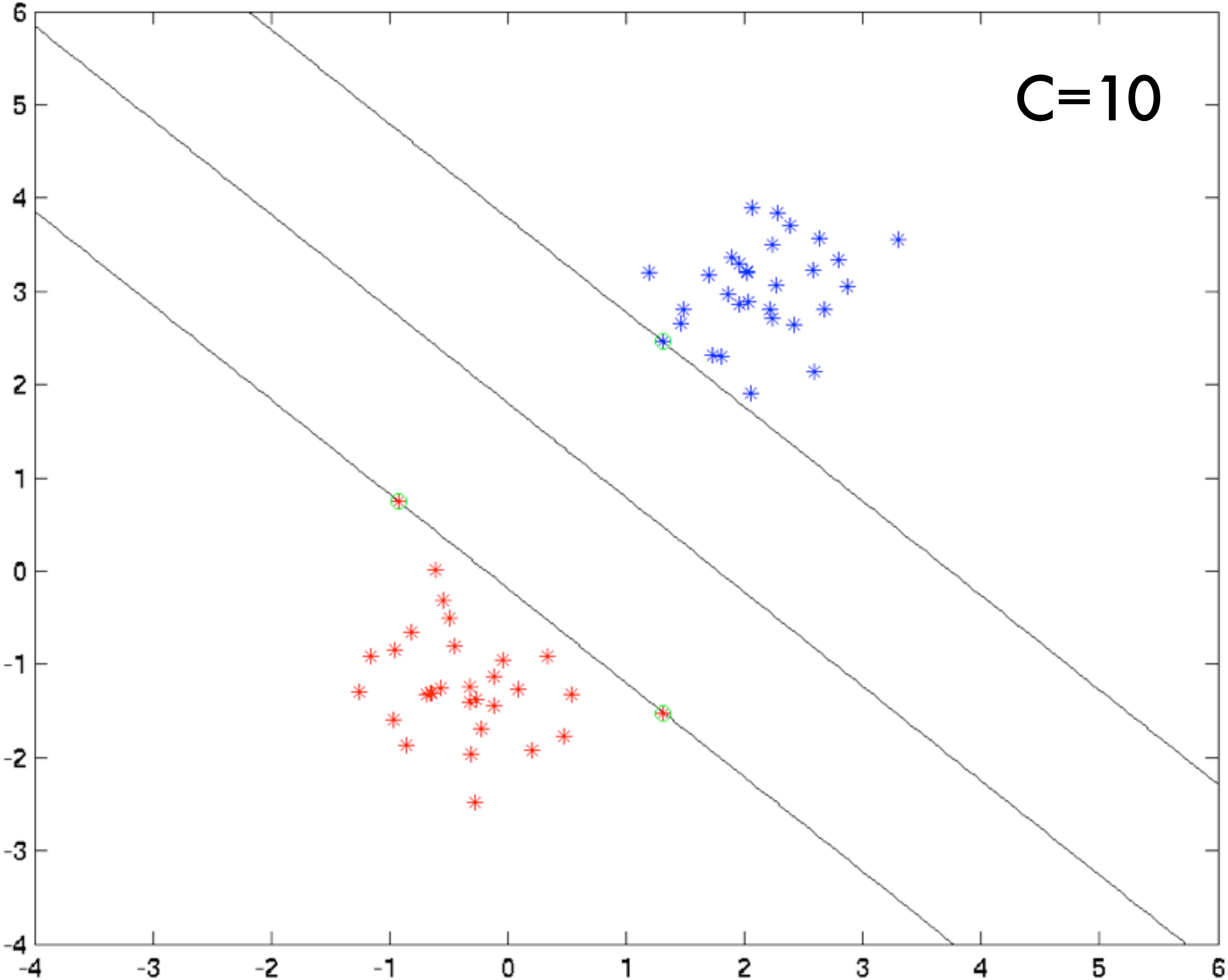
C=50
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

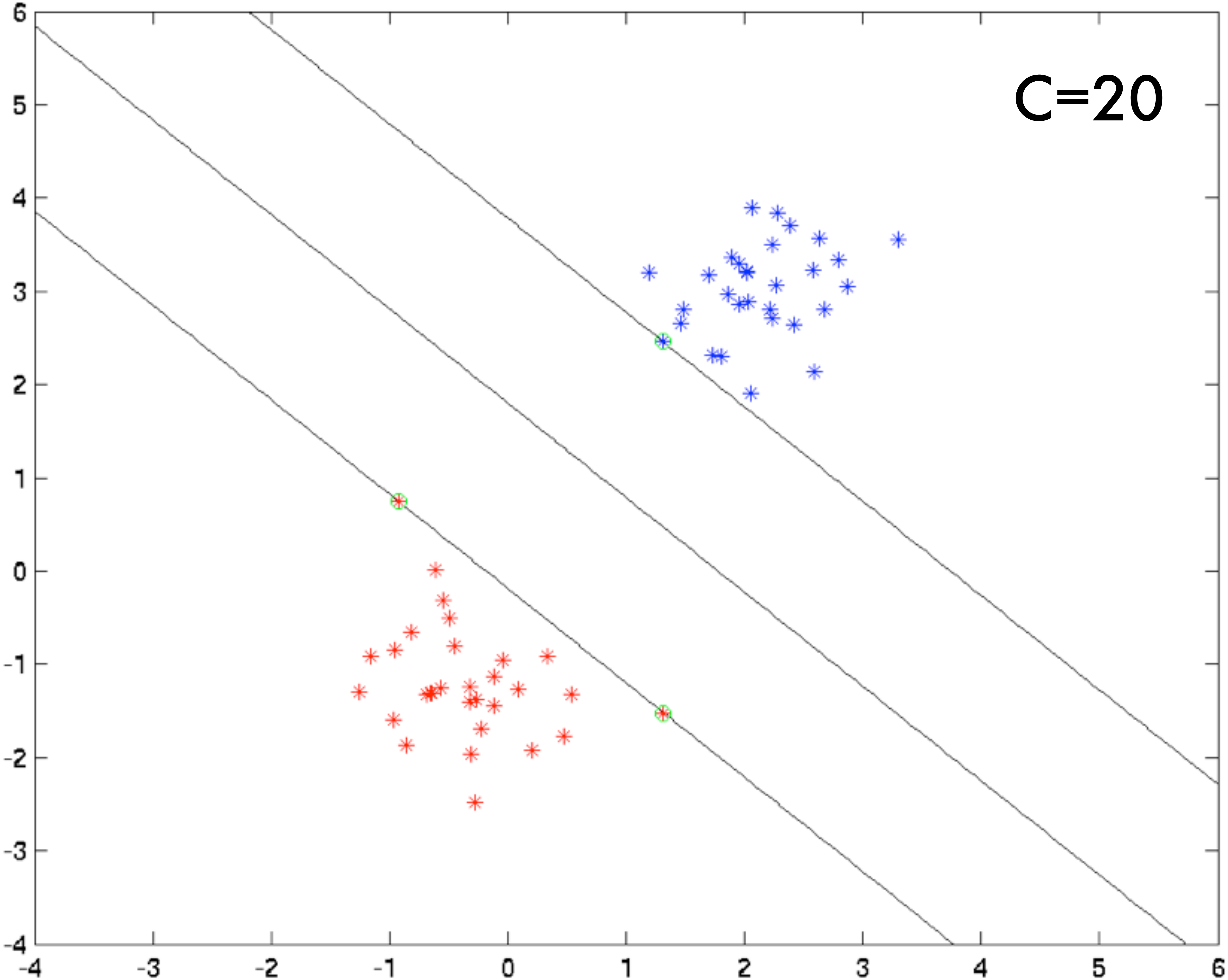

C=100
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6


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

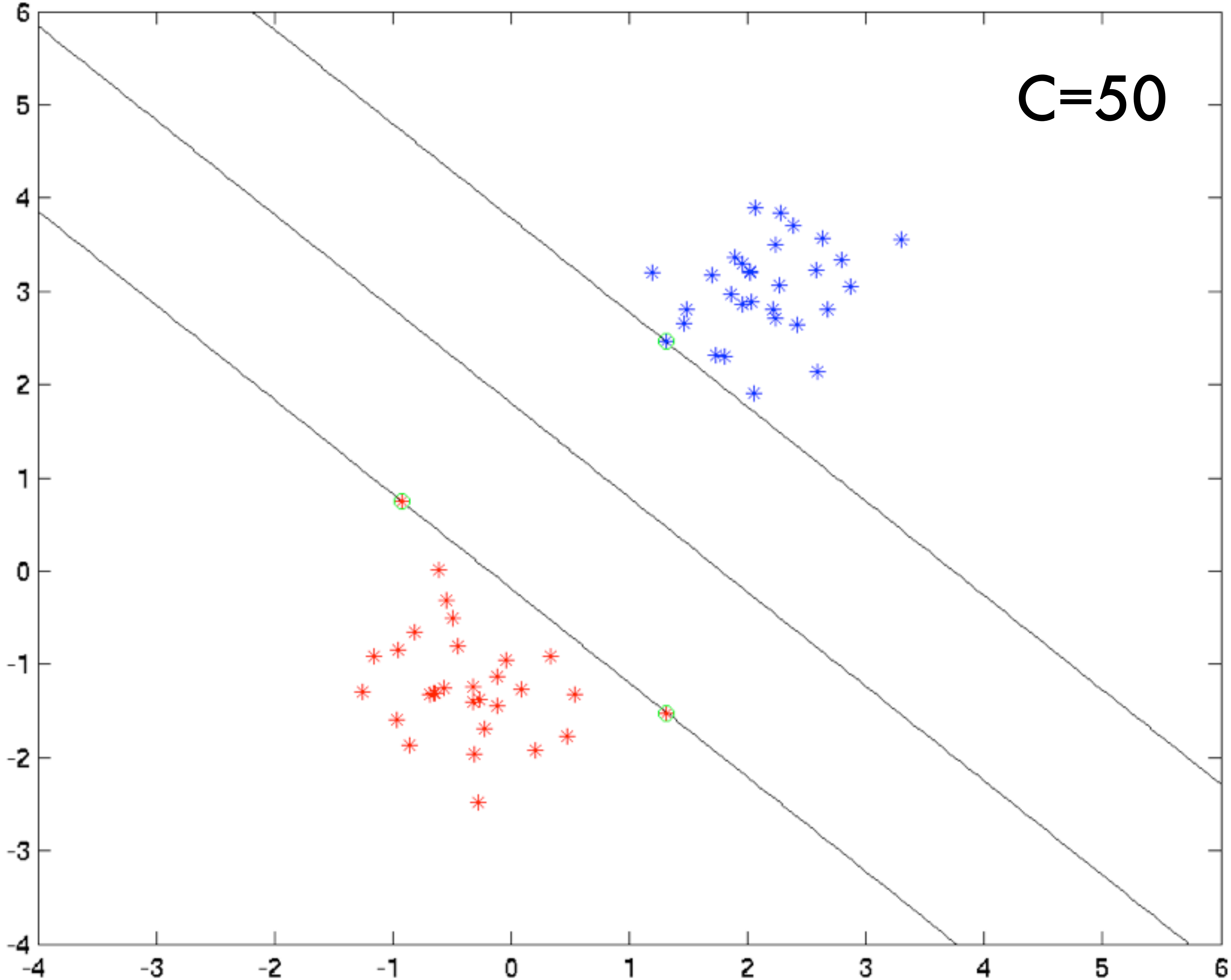

C=2
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6


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

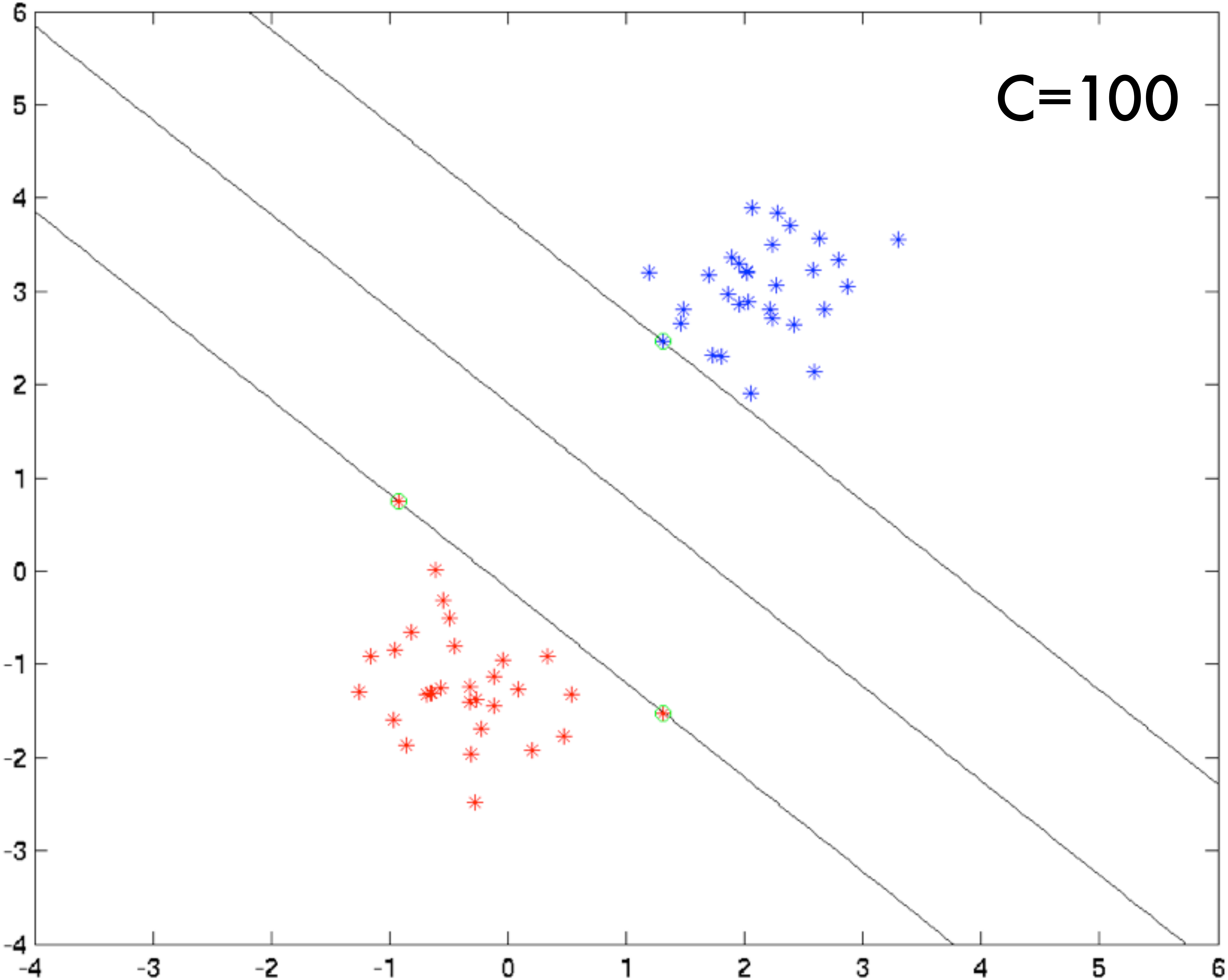

C=10
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

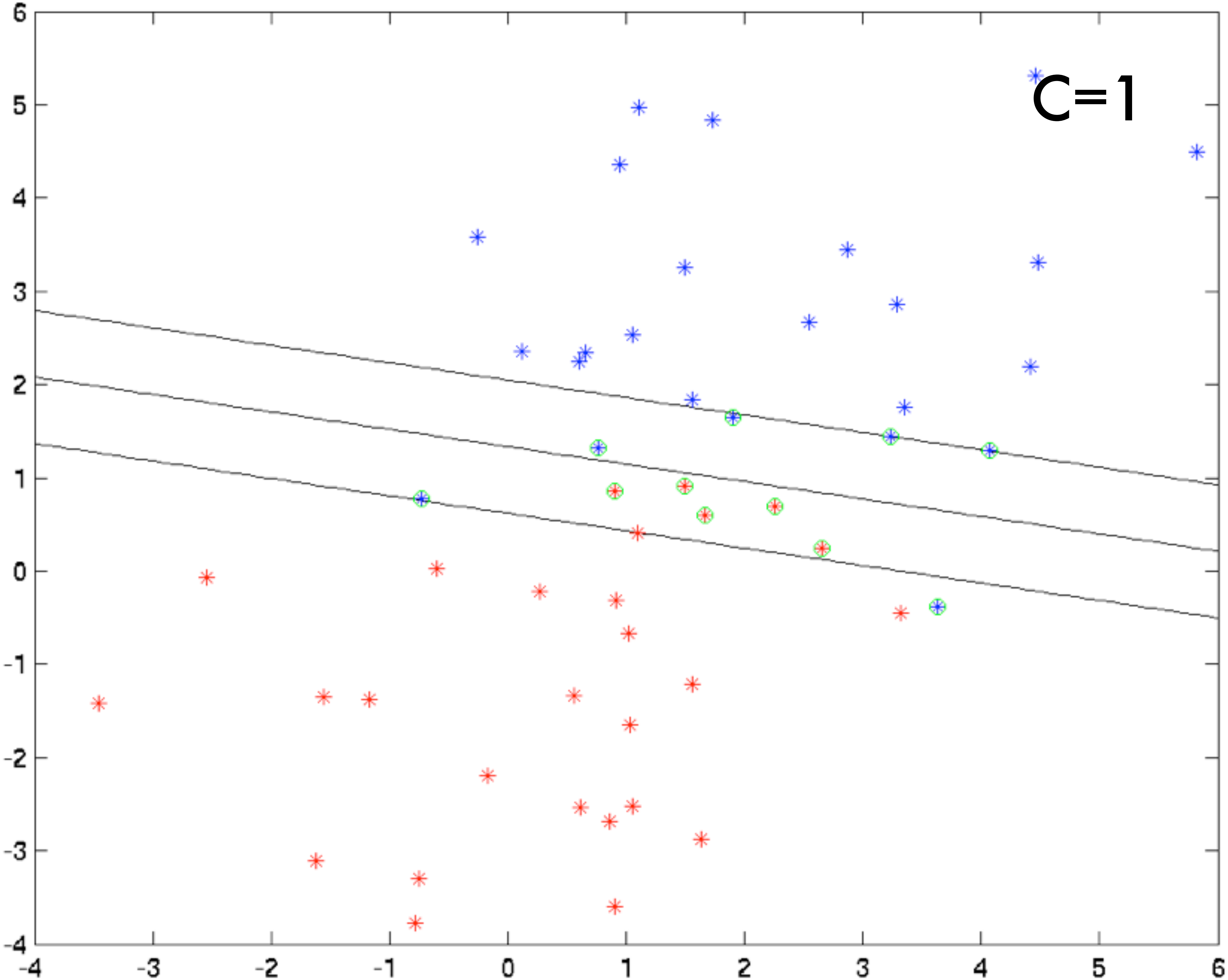


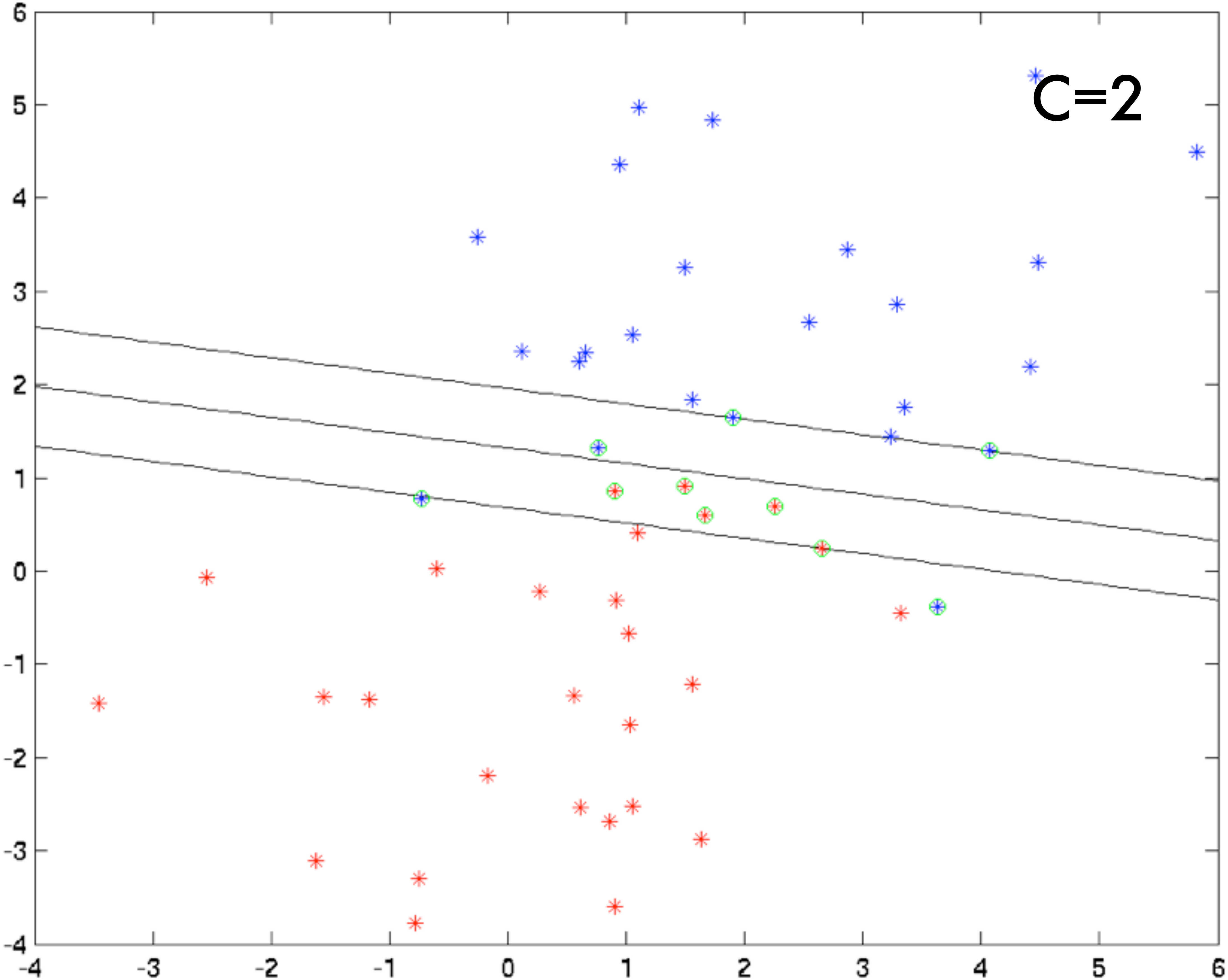
C=20

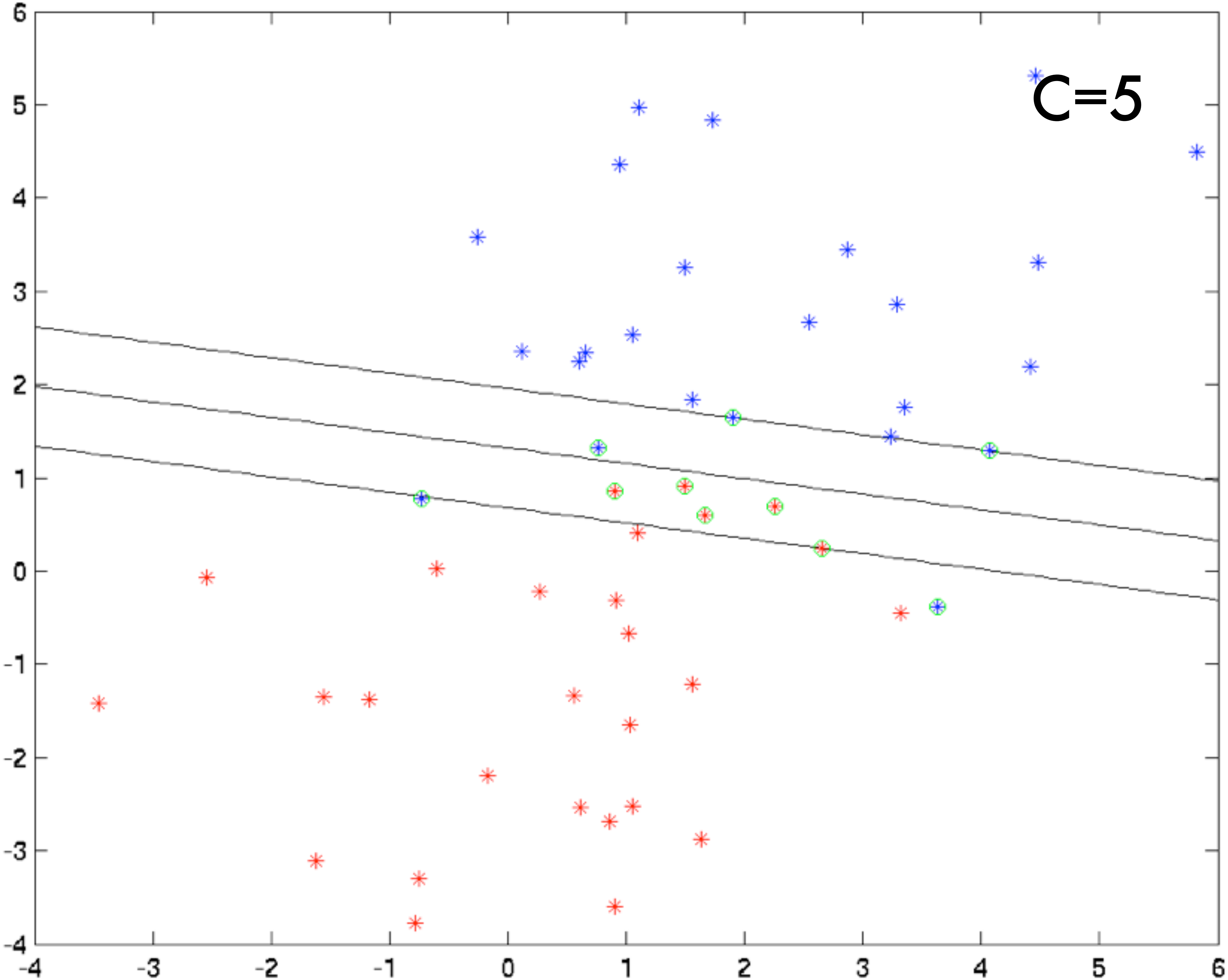


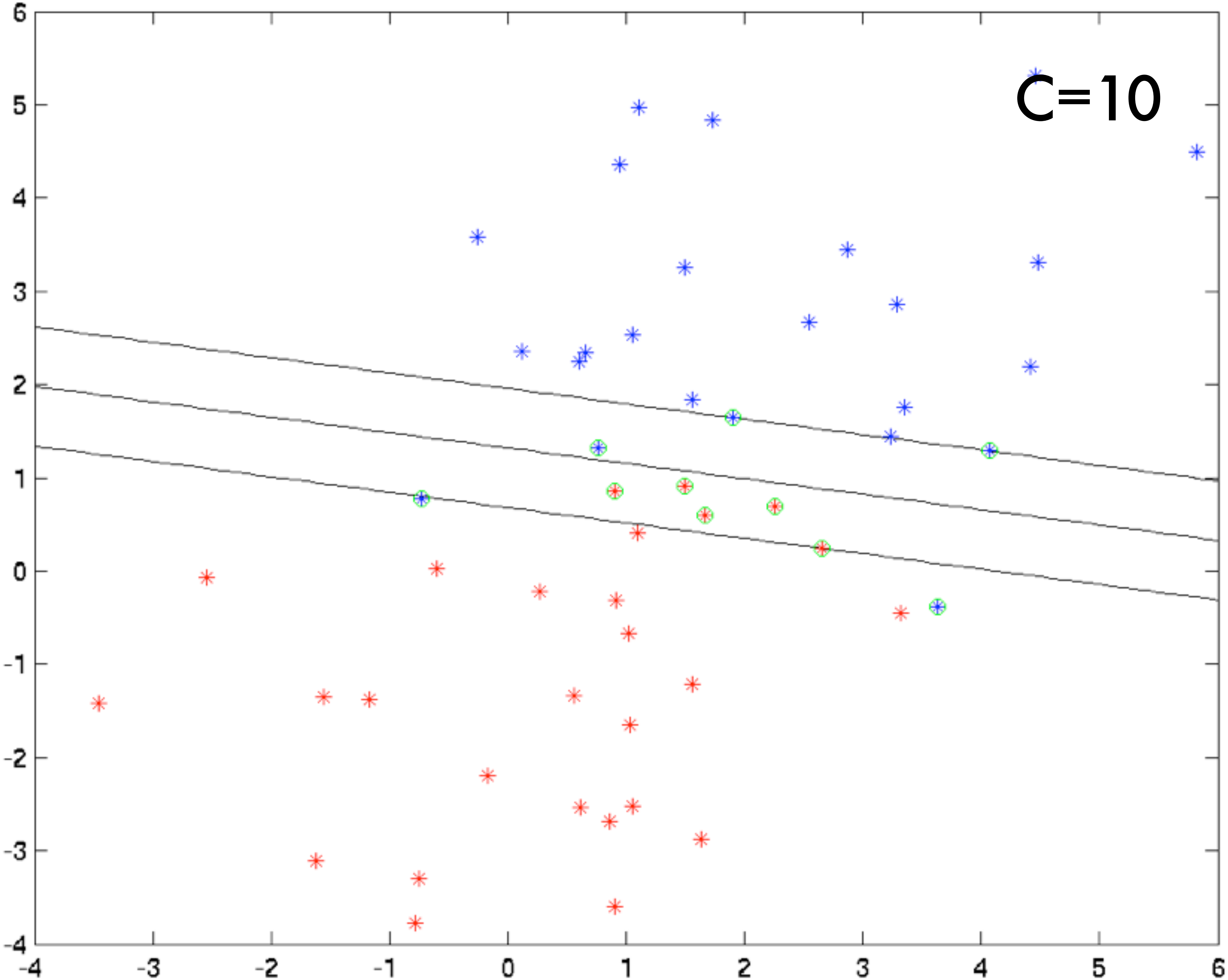
C=50
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

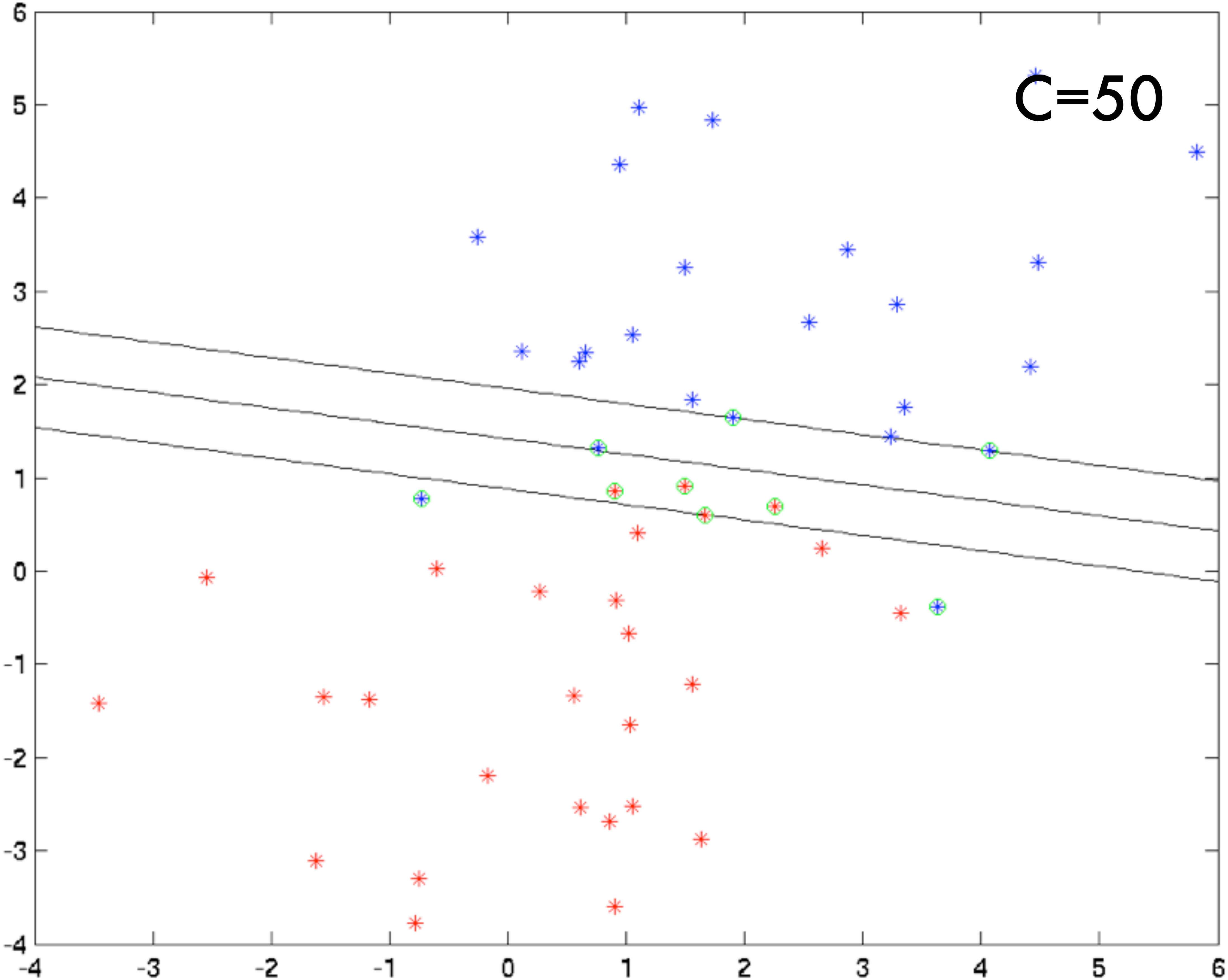

C=100
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6


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

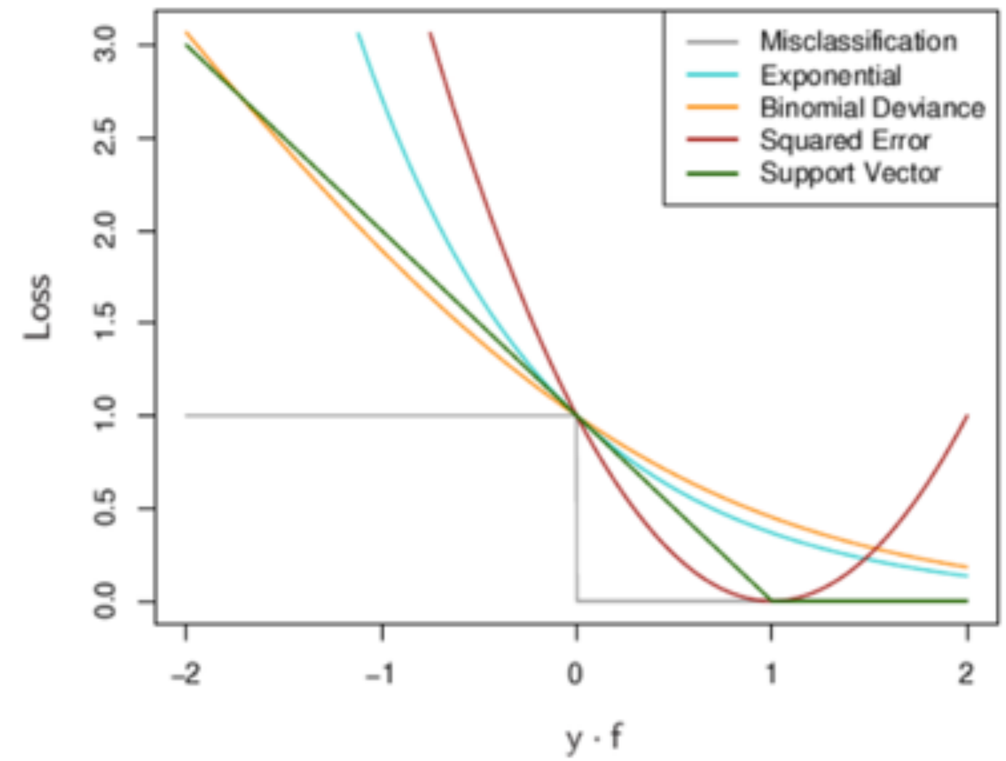

C=50
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4
5
6

Loss-function formulation

$$R_{\text{emp}}(w, w_0) := \sum_{i=1}^{N} \ell(y_i(w^T x_i + w_0))$$

Recall: predictor is

$$h(x) = \text{sgn}(w^T x + w_0)$$



[image: quora.com]

Good old NP-Hard formulation

$$\min_{w, w_0} \quad \tfrac{1}{2}\|w\|^2 + C \sum\nolimits_i [\![y_i \neq h(x_i)]\!]$$

Exercise: Look at our perceptron lecture and figure out what to do next!

Goal: `<irony>`ultimate goal of ML: apply SGD`</irony>`

Some other thoughts / ideas

* Novelty detection via SVMs (1-class SVM)
* SGD on hinge-loss SVM is actually very popular
* SVM history very inspiring: huge wave of ML exuberance
* How to obtain probabilities from SVM outputs?
* Bayesian SVMS (yep, people have tried that!)
* SVMs on hardware, low-power SVMs, etc.
* Can be a good choice for small to medium sized problems
* Can do test-time quite fast