Support Vector Machines

Justin Pounders

Introduction to SVM

Support Vector Machines (SVMs) are statistical models used for **classification**.

Review:
- What is classification?
- What other classification models have you seen?

How to Think about SVM

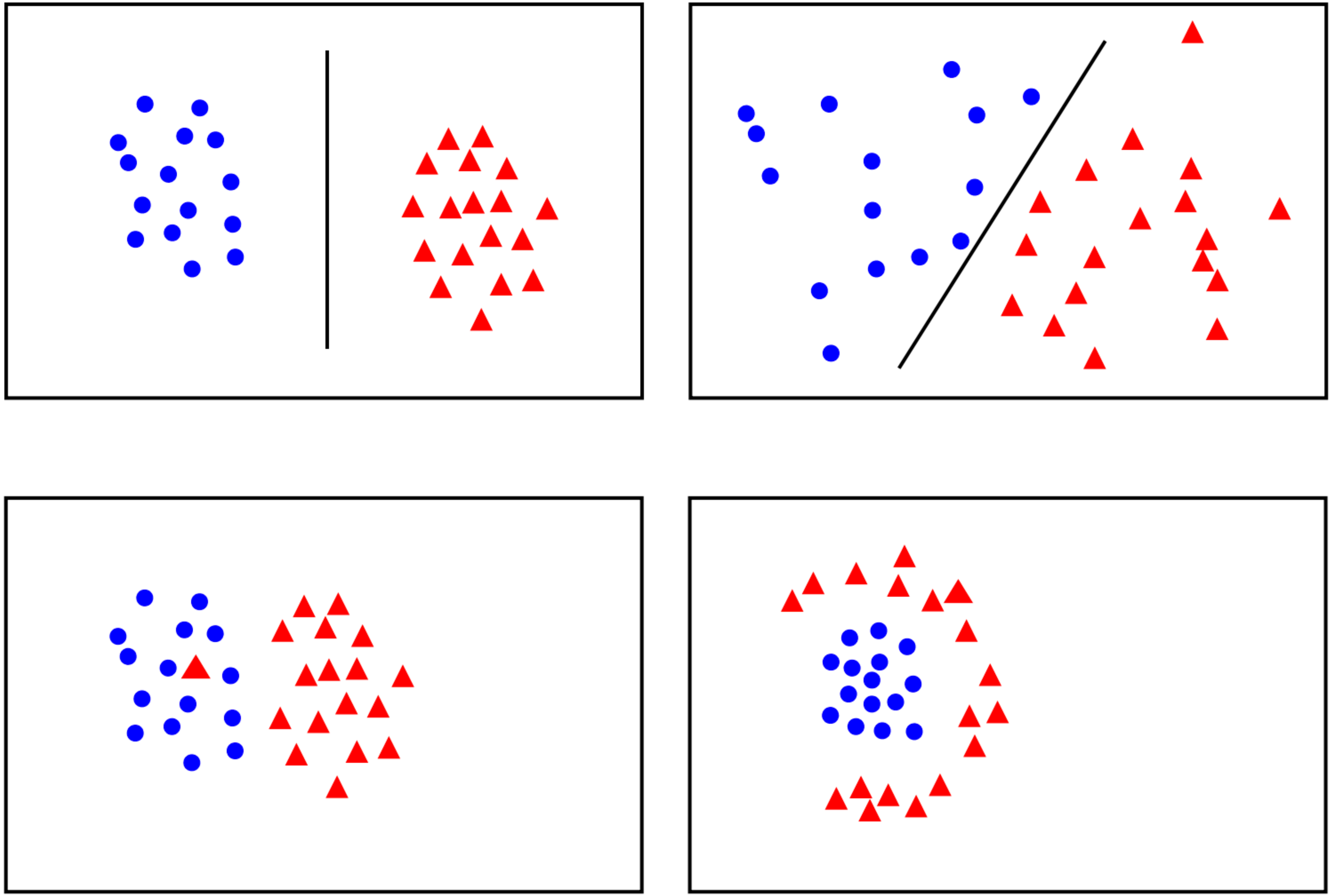
- The geometric intuition of SVMs is easier to grasp than the
- mathematical constructs needed to make it work

Linear Separability

SVMs work really well for data in which the classes are linearly separable.

linearly separable

not linearly separable



Maximum-Margin Estimator

If classes are linearly seperable, SVM finds the hyperplane that sepearates the classess with maximum margin.

— **What is a hyperplane?**

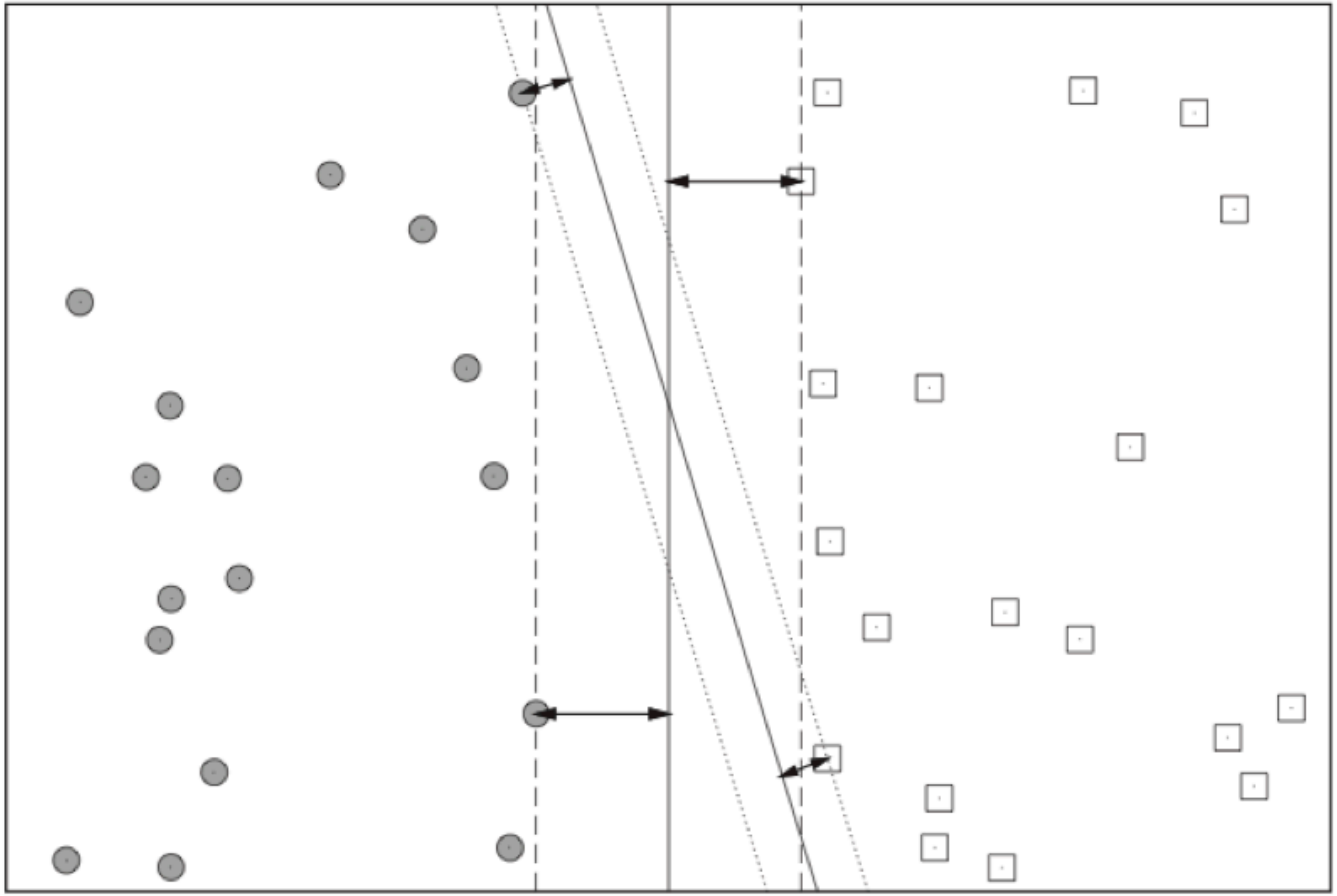
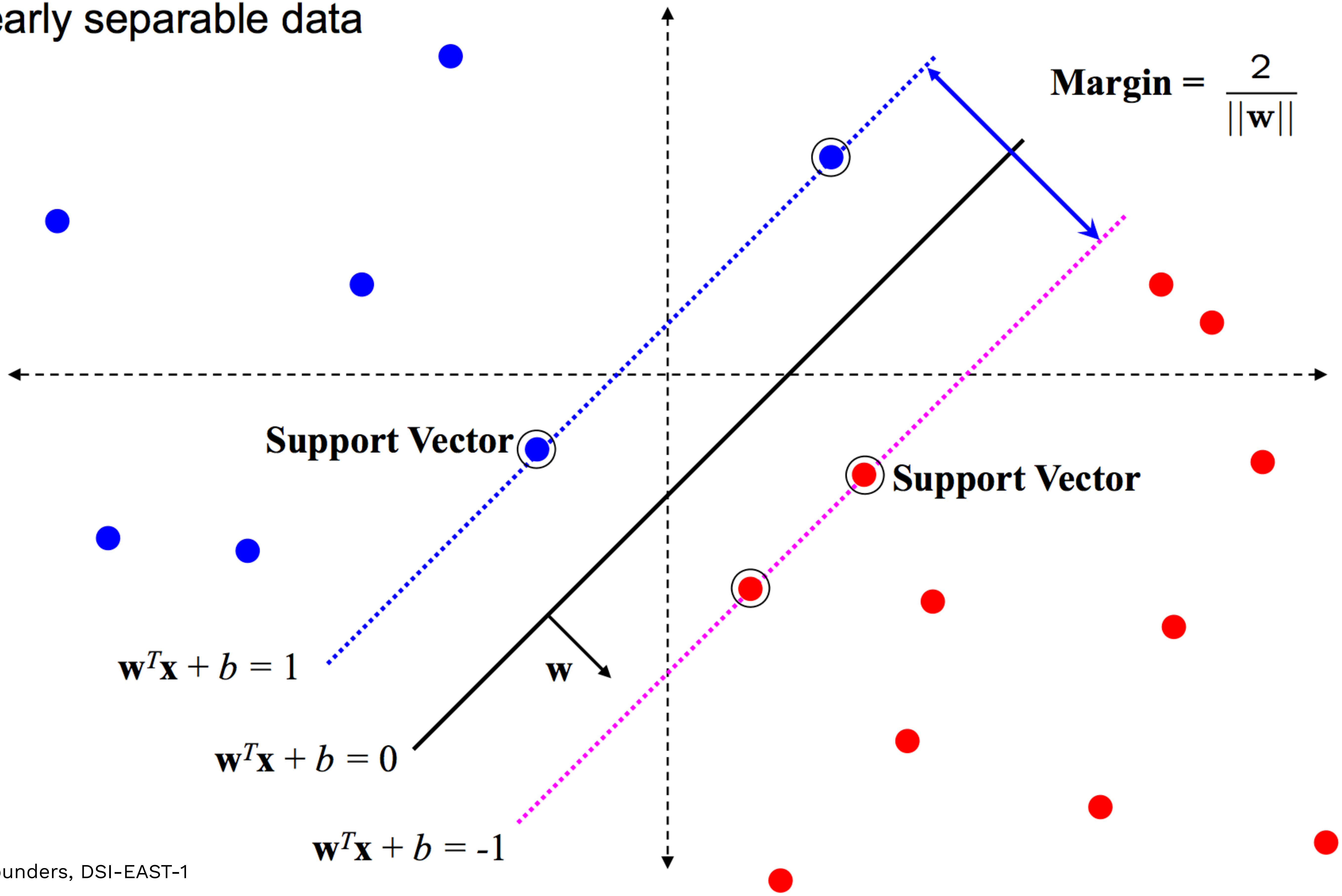


FIGURE 18-4. Two decision boundaries and their margins. Note that the vertical decision boundary has a wider margin than the other one. The arrows indicate the distance between the respective support vectors and the decision boundary.

Why maximize the margin?

- SVM solves for a decision boundary that should minimize the generalization error.
- Observations near the decision boundary are the most "ambiguous"
- SVM defines it's fit using the most ambiguous points

linearly separable data



Maximum-Margin Hyperplane

Goal: Find w that leads to the max-margin hyperplane

$$w \leftarrow \max_{w} \frac{2}{||w||} = \max \text{margin}$$

subject to all points being on the "right side"

$$\begin{aligned} w^T x_i + b &\geq 1 \quad \text{if } y_i = 1 \\ w^T x_i + b &\leq -1 \quad \text{if } y_i = -1 \end{aligned}$$

Maximum-Margin Hyperplane

What if data are not linearly seperable?

Maximum-Margin Hyperplane

What if data are not linearly seperable?

— Still want to minimize $\|w\|$ (maximize margin)

Maximum-Margin Hyperplane

What if data are not linearly seperable?

- Still want to minimize $\|w\|$ (maximize margin)
- Would also like to minimize a loss function that penalizes points for being on the "wrong side"

Hinge Loss Function

$$\text{hinge loss} = \sum_{i=1}^{n} \max\left[0, 1 - y_i(w^T x_i + b)\right]$$
$$= \begin{cases} 0 & \text{if } x \text{ outside or on margin} \\ > 0 & \text{if } x \text{ within margin} \end{cases}$$

Hinge loss penalizes misclassified points!

Maximum Margin Hyperplane

Put "simply" want to minimize

$$C \times \left[\text{hinge loss}\right] + \left[\frac{1}{\text{margin width}}\right]$$

where C is a hyperparameter.

Maximum Margin Hyperplane

Put "simply" want to minimize

$$[\text{hinge loss}] + \frac{1}{C}\left[\frac{1}{\text{margin width}}\right]$$

$$\sum_{i=1}^{N} \max\left(0, 1 - y_i(w^T x_i + b)\right) + \frac{1}{C}||w||^2$$

i.e., loss + 1/C * regularization (c.f. Ridge!)

Maximum Margin Hyperplane

Takeaway: **Bias/variance trade-off** is handled via the hyperparameter C

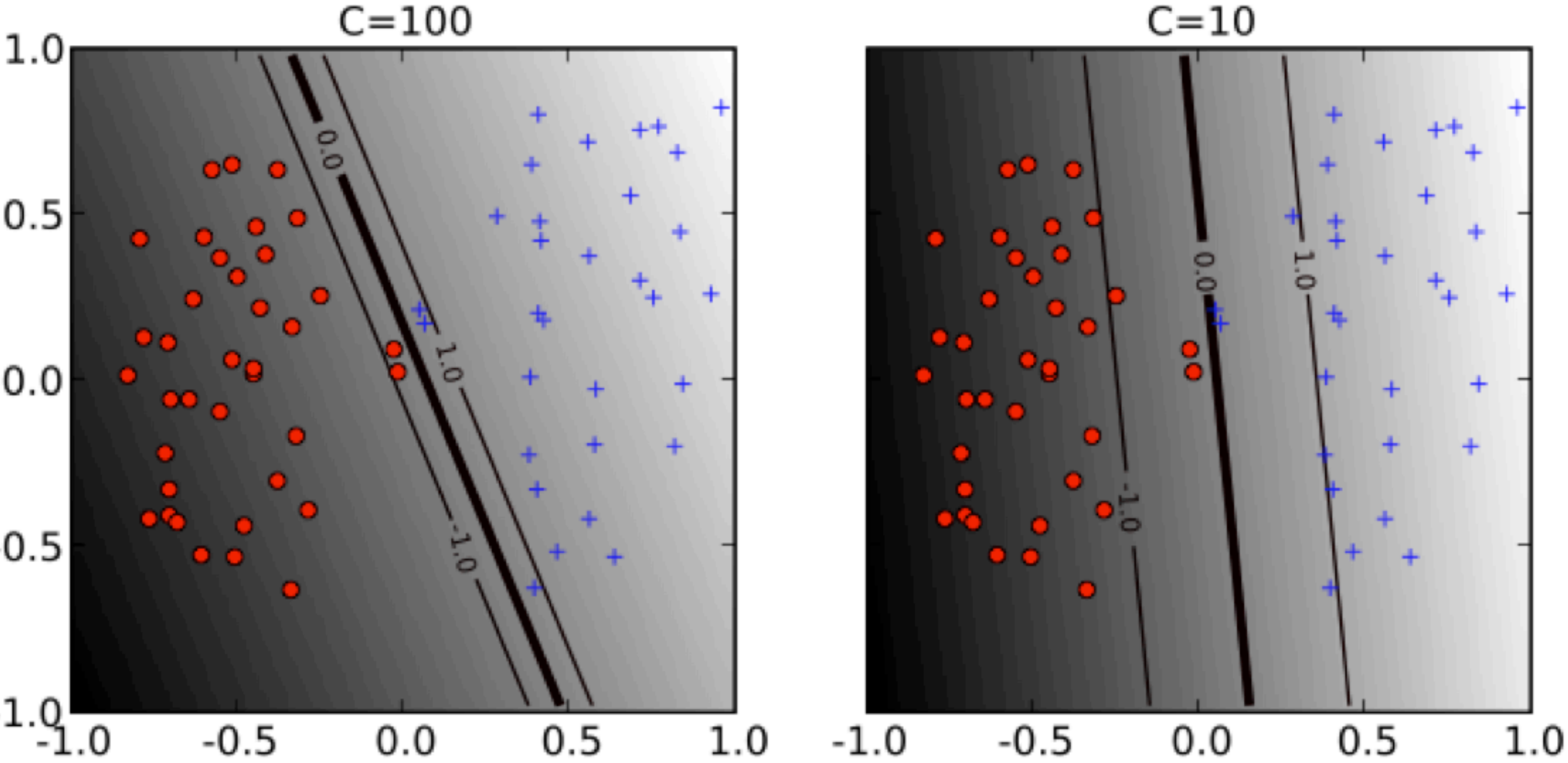
$$C \times [\text{hinge loss}] + \left[\frac{1}{\text{margin width}}\right]$$

Maximum Margin Hyperplane

$$C \times [\text{hinge loss}] + \left[\frac{1}{\text{margin width}}\right]$$

— Large $C \to$ narrow margin, less tolerant of misclassification, tends toward high variance

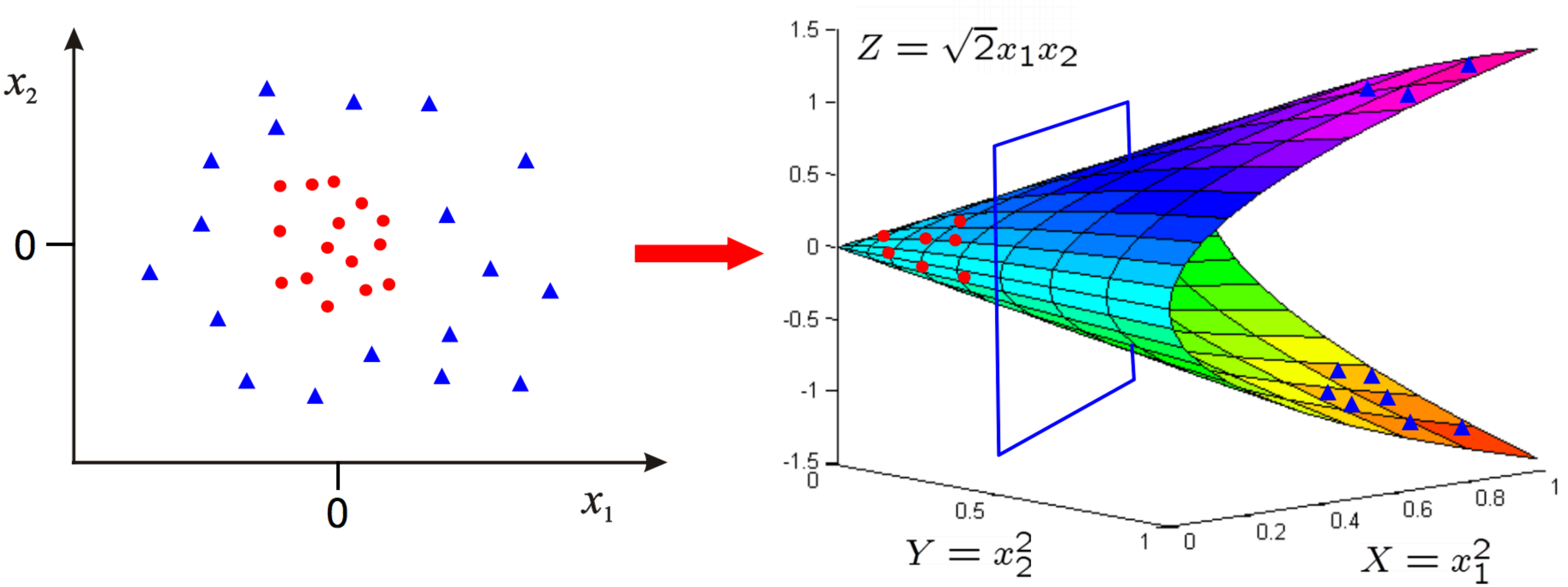
— Small $C \to$ wider margin, more tolerant of misclassification, tends toward high bias


C=100
C=10
1.0
0.5
0.0
0.5
1.0
-1.0
-0.5
0.0
0.5
1.0
0.0
1.0
-1.0

What if your data are not separable?

Like, no where close to linearly separable?

$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \qquad \mathbb{R}^2 \rightarrow \mathbb{R}^3$$



- Data is linearly separable in 3D
- This means that the problem can still be solved by a linear classifier

What if you data are not separable?

Kernel trick:

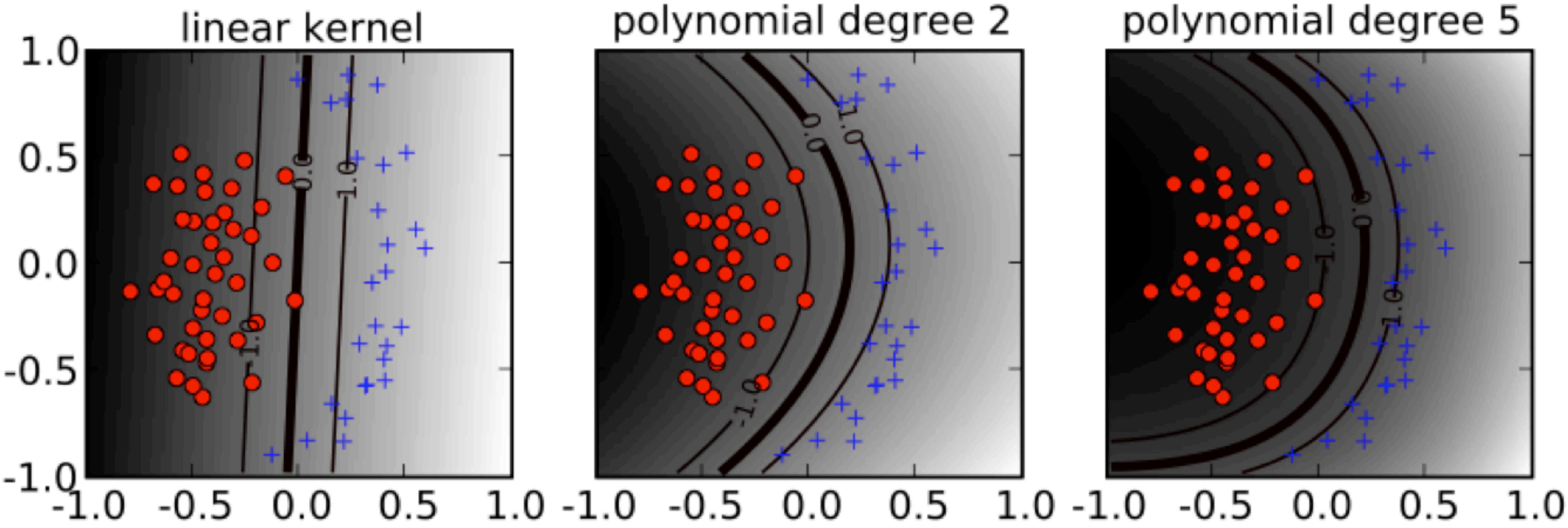
Replace

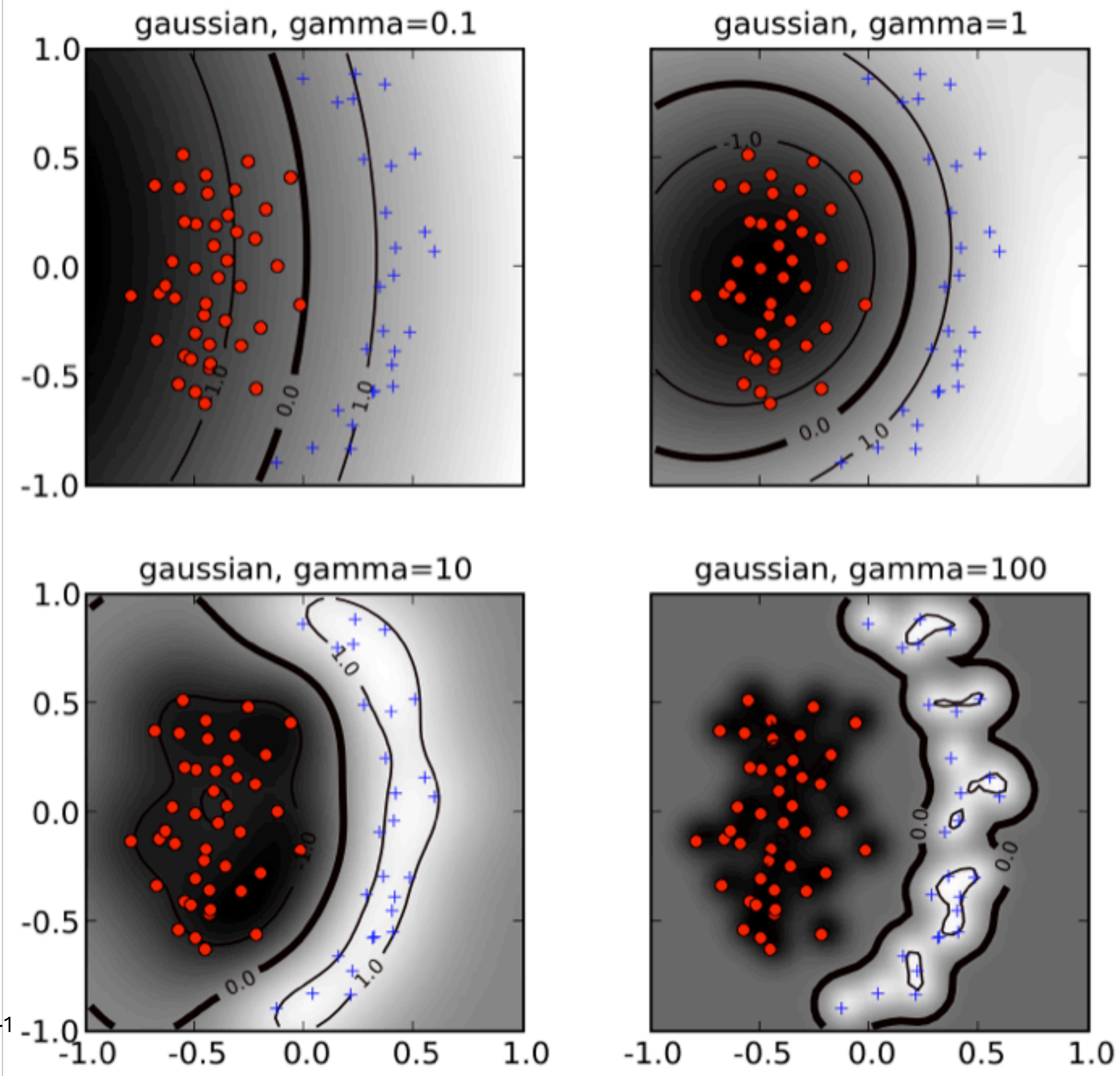
$$x \leftarrow \Phi(x)$$

$$w \leftarrow \Phi(w)$$

Kernel SVM

- Linear
- Polynomial
- (Gaussian) Radial Bassis Function (RBF)


linear kernel
polynomial degree 2
polynomial degree 5
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.5
0.0
0.5
1.0
-1.0
0.0
1.0



gaussian, gamma=0.1
gaussian, gamma=1
gaussian, gamma=10
gaussian, gamma=100

Pros and cons

Pros

- Exceptional perfomance (historically widely used)
- Robust to outliers
- Effective in high dimensional data
- Can work with non-linearities
- Fast to compute even on non-linear (kernel trick)
- Low risk of overfitting

Pros and cons

Cons
- Blackbox
- Can be slow on large datasets

When to use SVM vs. Logistic Regression

Advice from Andrew Ng:

If there are more feature than training samples:
Use logistic regression or SVM without a kernel ("linear kernel")

If there are about 10 times as many samples as features:
Use SVM with a Gaussian kernel

If there are many more training samples than features:
Spend time feature engineering, then use logistic regression or SVM without a kernel

Additional resources

Taken from `introduction-to-svm.ipynb` **in rep.**

- For a really great resource check out these slides (some of which are cannabalized in this lecture).
- This website is also a great resource, on a slightly more technical level.
- SVM docs on SKLearn
- Iris example on SKLearn
- Hyperplane walkthrough on SKLearn
- A comprehensive user guide to SVM. My fav!

Additional resources

Taken from `introduction-to-svm.ipynb` in rep.

- A blog post tutorial of understanding the linear algebra behind SVM hyperplanes. Check part 3 of this blog on finding the optimal hyperplane
- This Quora discussion includes a high-level overview plus a 20min video walking through the core "need-to-knows"
- A slideshow introduction to the optimization considerations of SVM

Additional resources

Taken from `introduction-to-svm.ipynb` **in rep.**

- Andrew Ng's notes on SVM from CS 229
- A FULL LECTURE (1hr+) from one of my fav lecturers (Dr Yasser) on SVM. He does a followup on kernel tricks too
- A FULL LECTURE (50min) (from MIT Opencoursewar)
- An infamous paper (cited 7000+ times!) on why SVM is a great text classifier
- An advanced discussion of SVMs as probabilistic