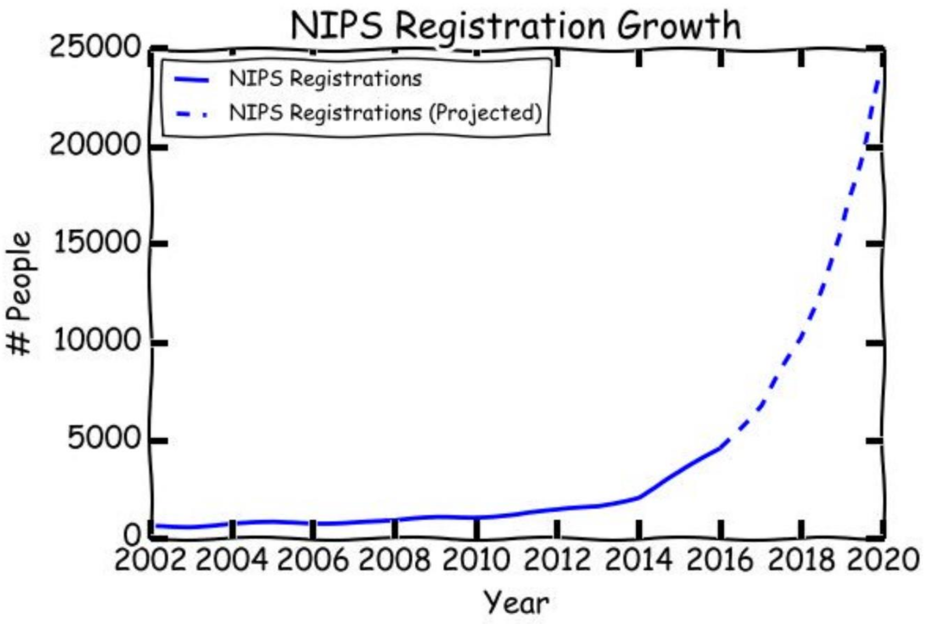
Support Vector Machines

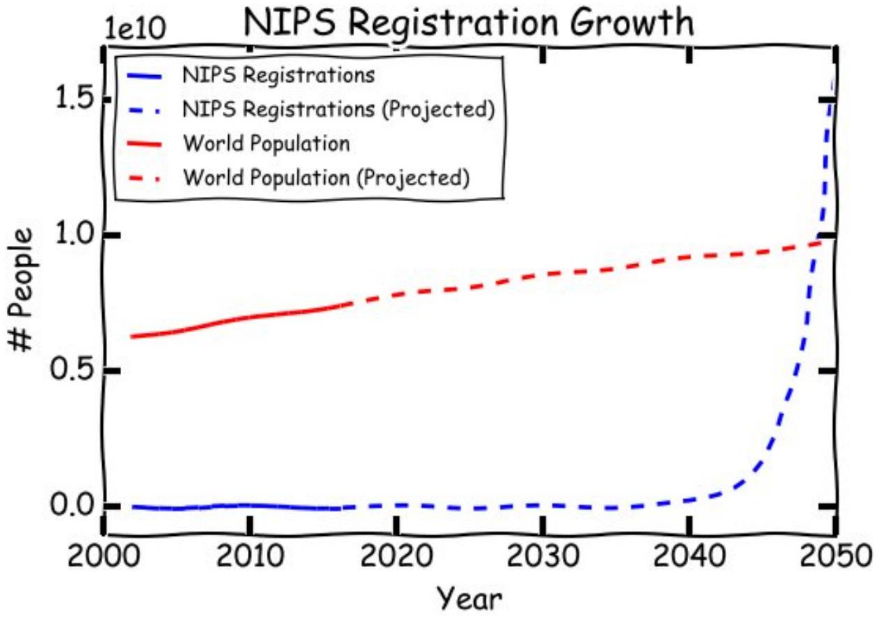
ddebarr@uw.edu

2017-03-02

ML Hipster
@ML_Hipster

By my calculations, the ML singularity will arrive around 2048.





Course Outline

1. Introduction to Statistical Learning
2. Linear Regression
3. Classification
4. Resampling Methods
5. Linear Model Selection and Regularization
6. Moving Beyond Linearity
7. Tree-Based Methods
8. Support Vector Machines
9. Unsupervised Learning
10. Neural Networks and Genetic Algorithms

Agenda

The Classification Setting

$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}, \ldots, x_n = \begin{pmatrix} x_{n1} \\ \vdots \\ x_{np} \end{pmatrix}$$

$$y_1, \ldots, y_n \in \{-1, 1\}$$

The Hyperplane

- In two dimensions:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$$

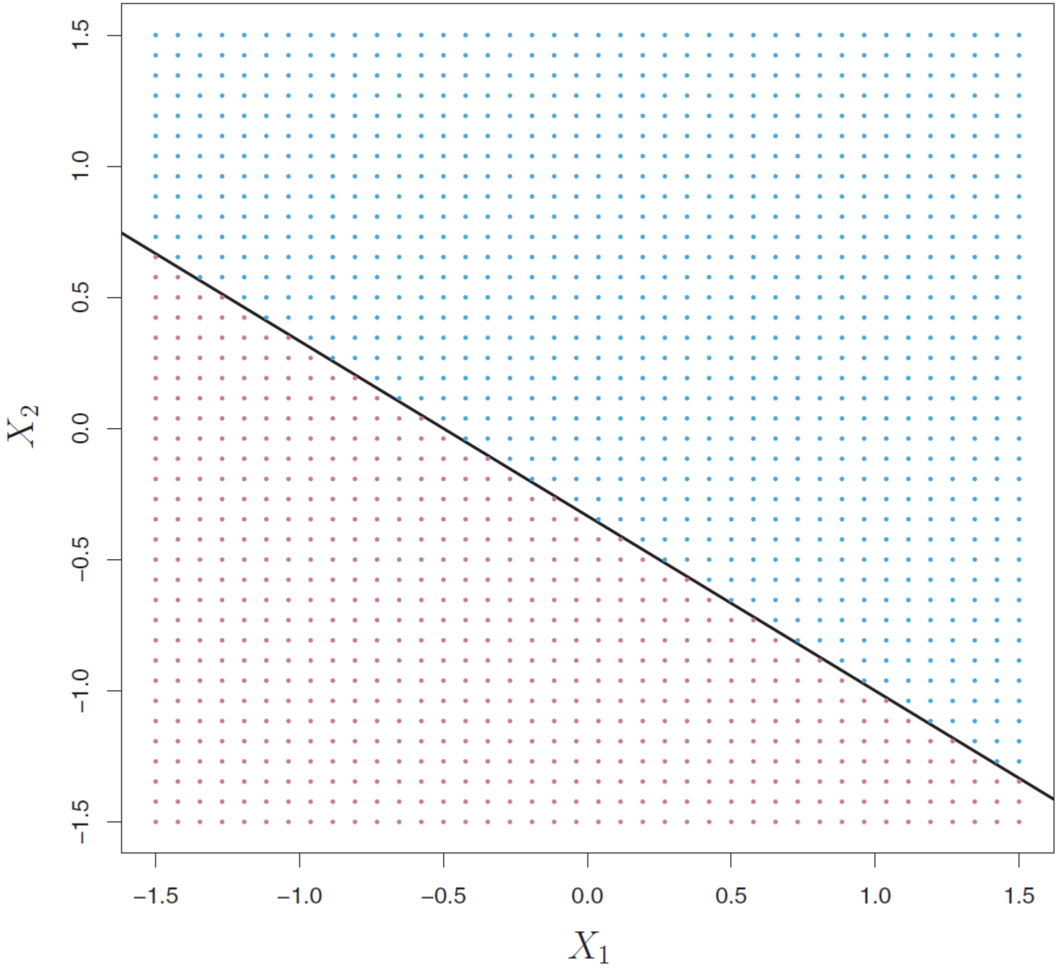
- In "p" dimensions:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$$

- Classify as "positive" if

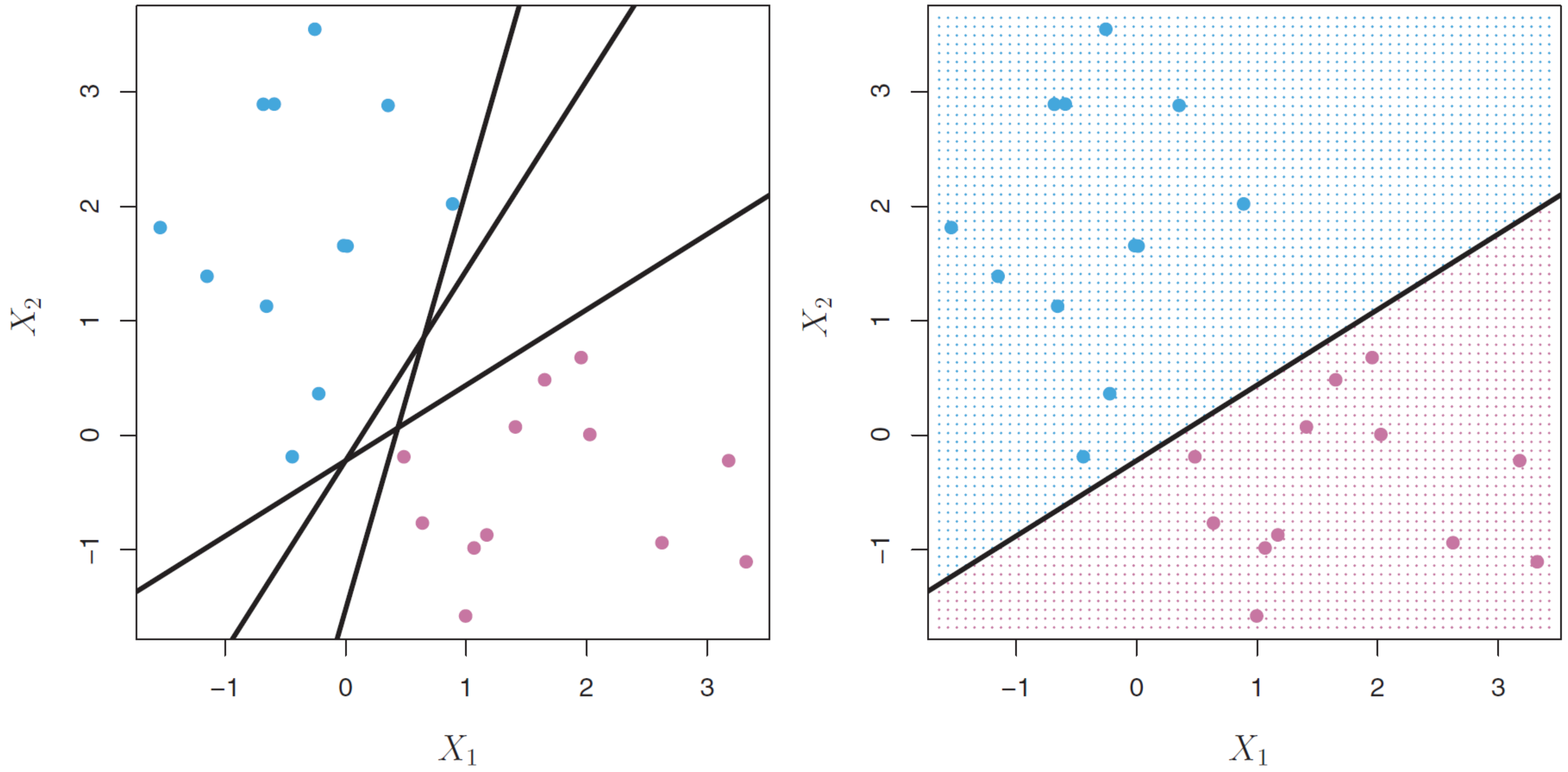
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p > 0$$

“Hyperplane” Example



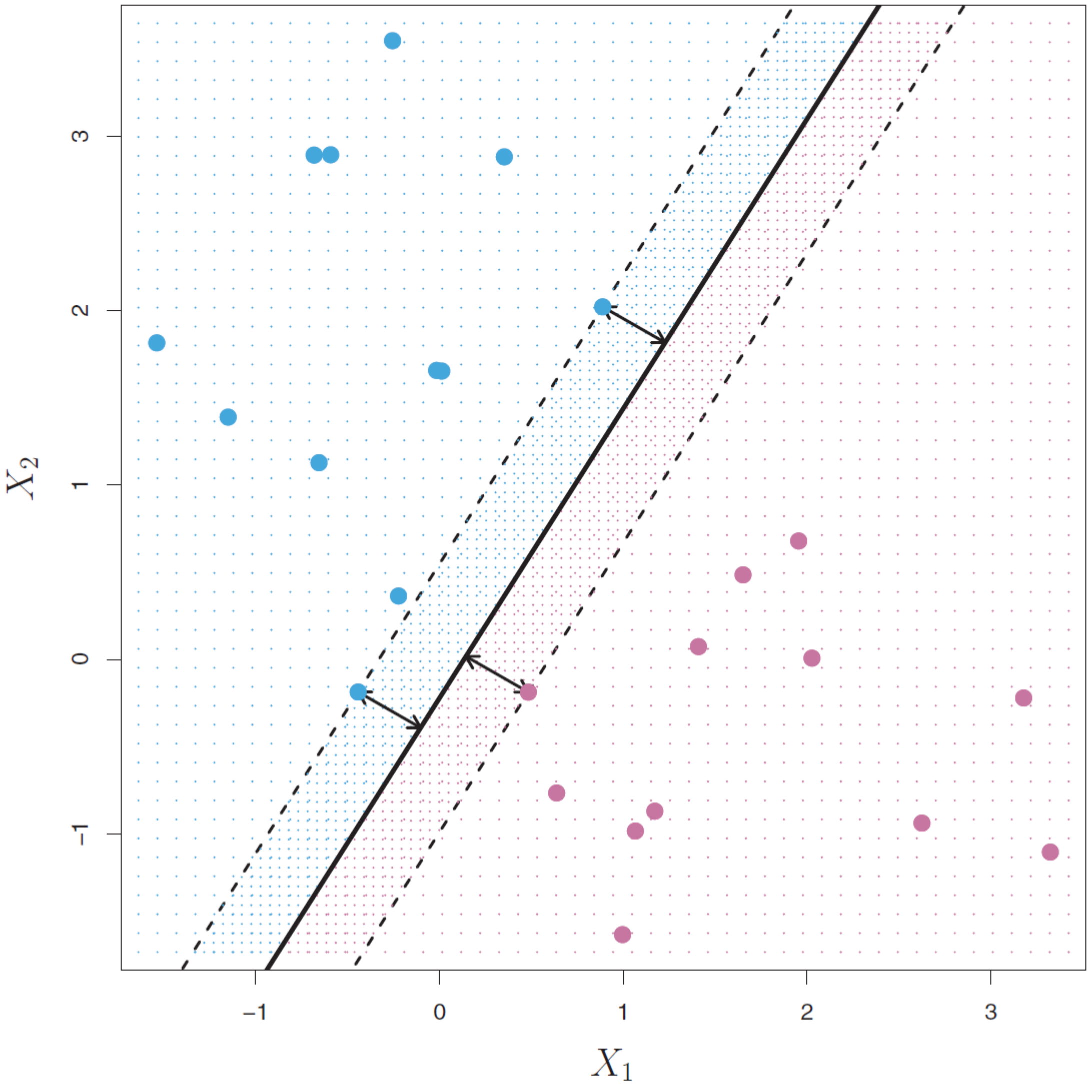
$1 + 2 * X_1 + 3 * X_2 = 0$

Separating Hyperplanes



Maximum Margin Hyperplane

- Margin: the minimal distance from the observations to the separating hyperplane
- We'd like to maximize this
- The observations nearest the decision boundary are known as support vectors, because they "support" (define) the decision boundary



The Maximal Margin Hyperplane Optimization Problem

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\, M$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

Note that index *j* starts at 1 for this constraint

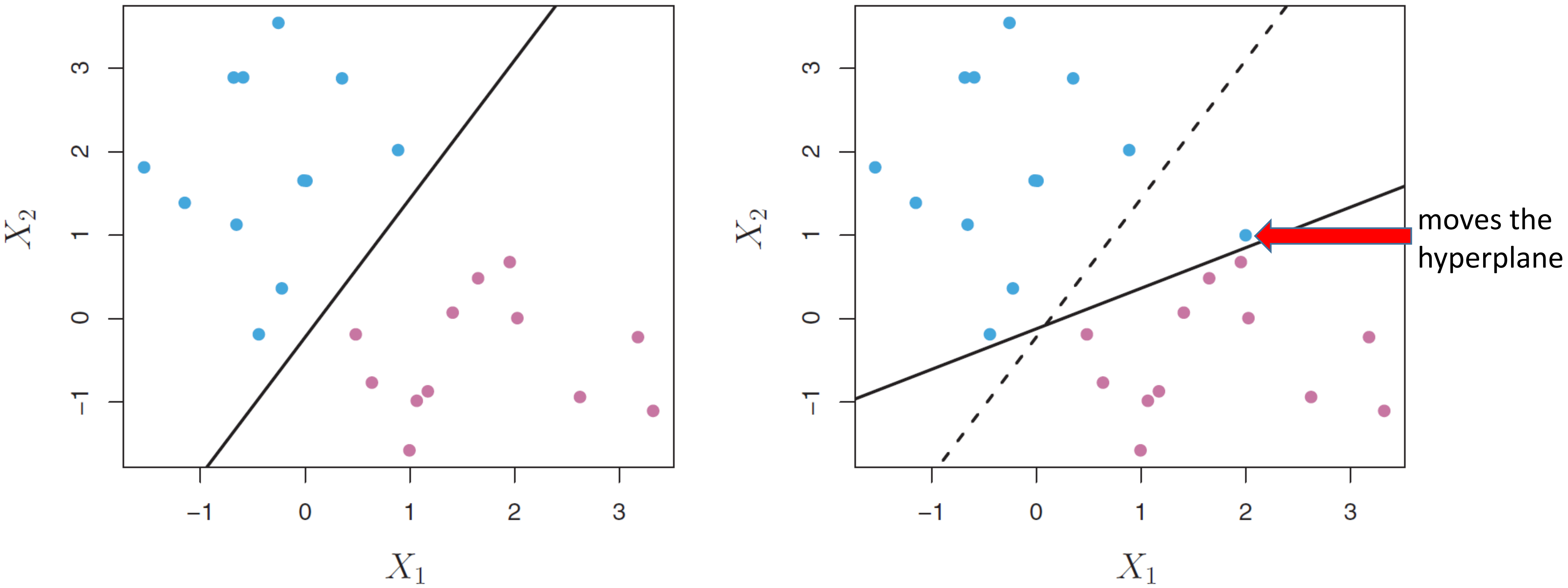
$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M \;\; \forall\, i = 1, \ldots, n$$

That last constraint requires that observations lie on the correct side of the hyperplane

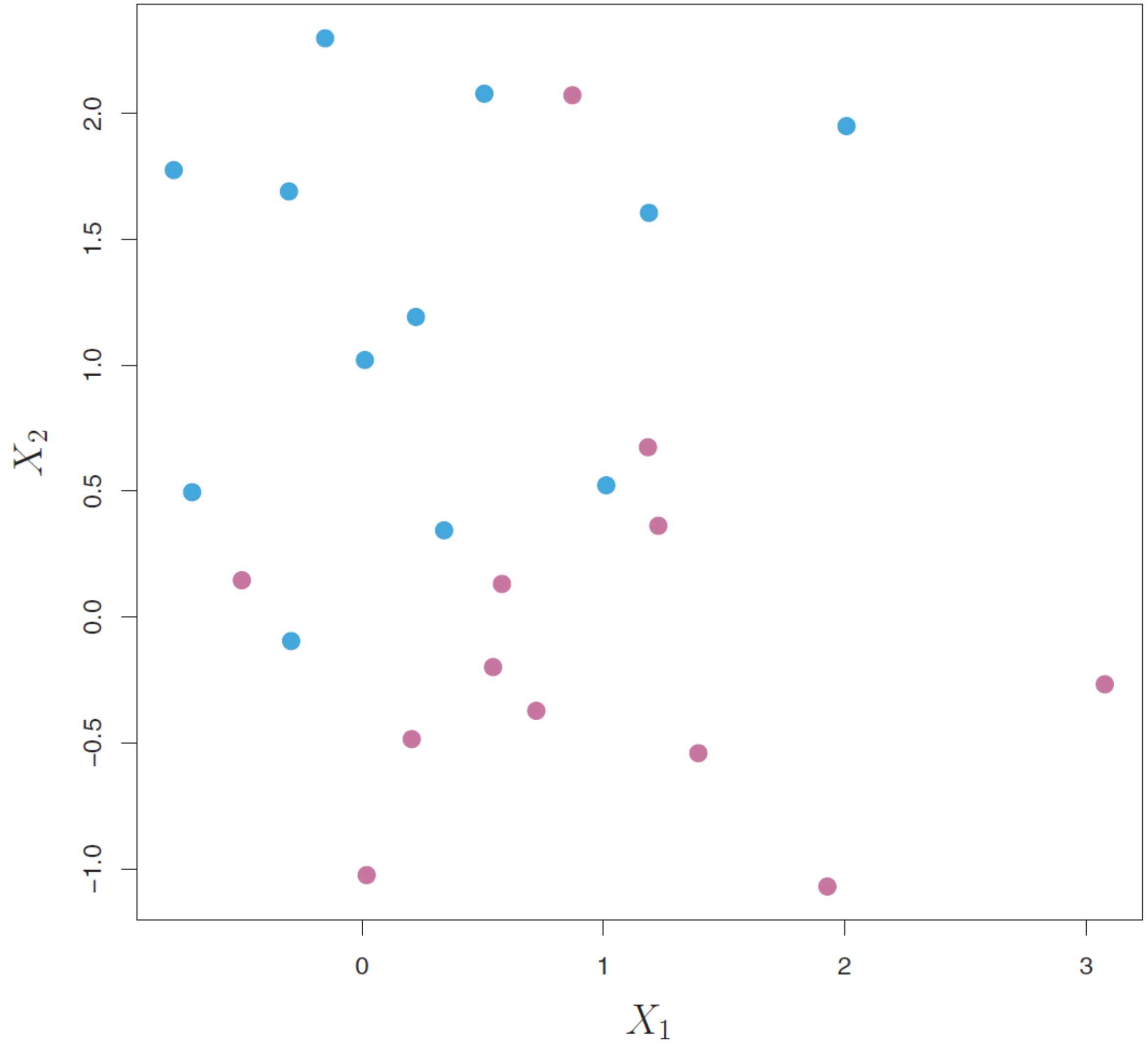
… which will not always be possible.

This is sometimes called a hard margin classifier, because errors are not allowed.

Maximal Margin Hyperplane is Sensitive to Small Changes in the Data



Classification Problem: Not Linearly Separable (linear model okay)



Consider using shape as well as color to distinguish between the positive and negative classes

(e.g. “x” and “o”, or “+” and “-”)

The Support Vector Classifier

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

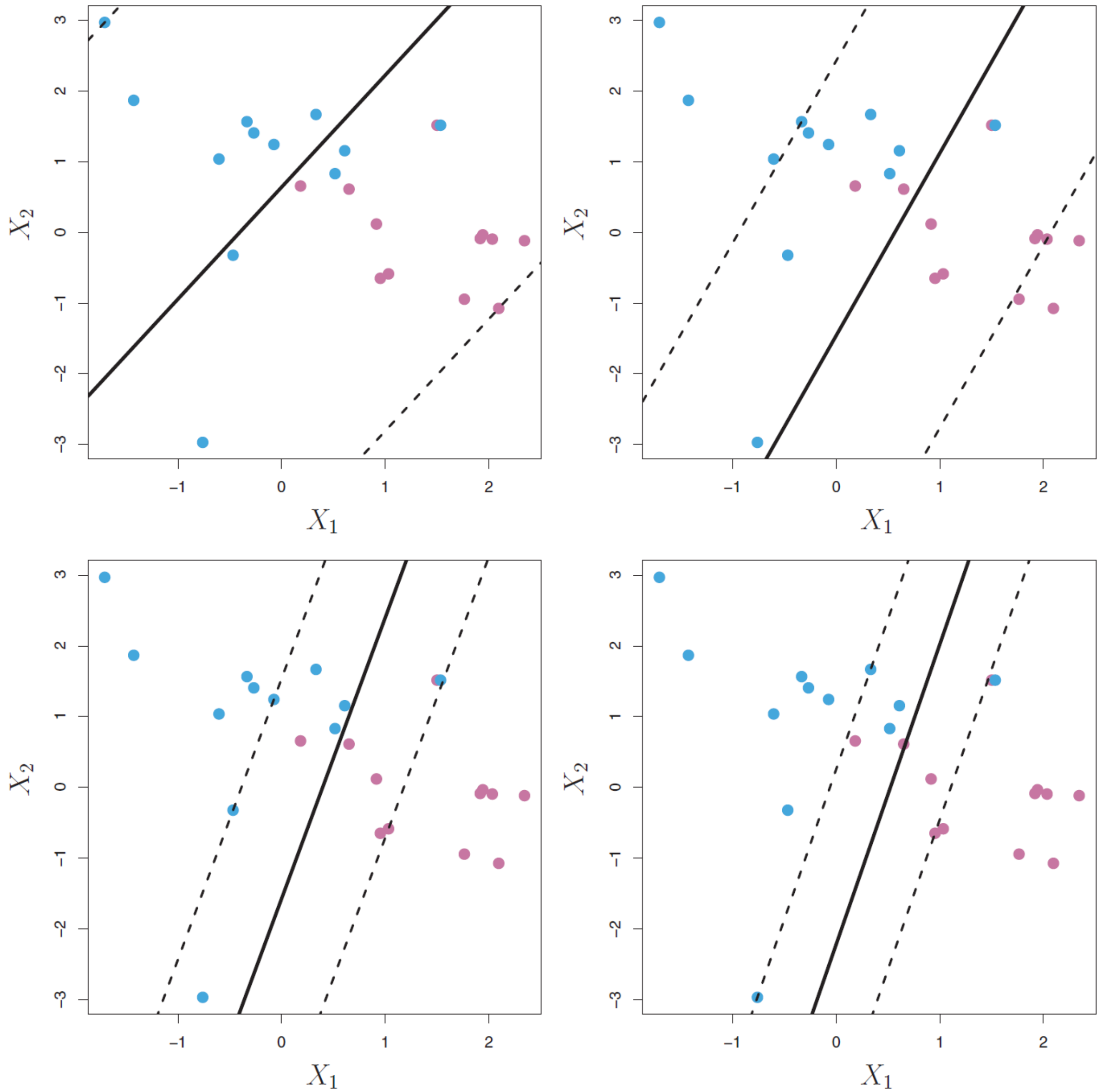
$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,$$

Based on hinge loss: $\max\left[0, 1 - y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})\right]$

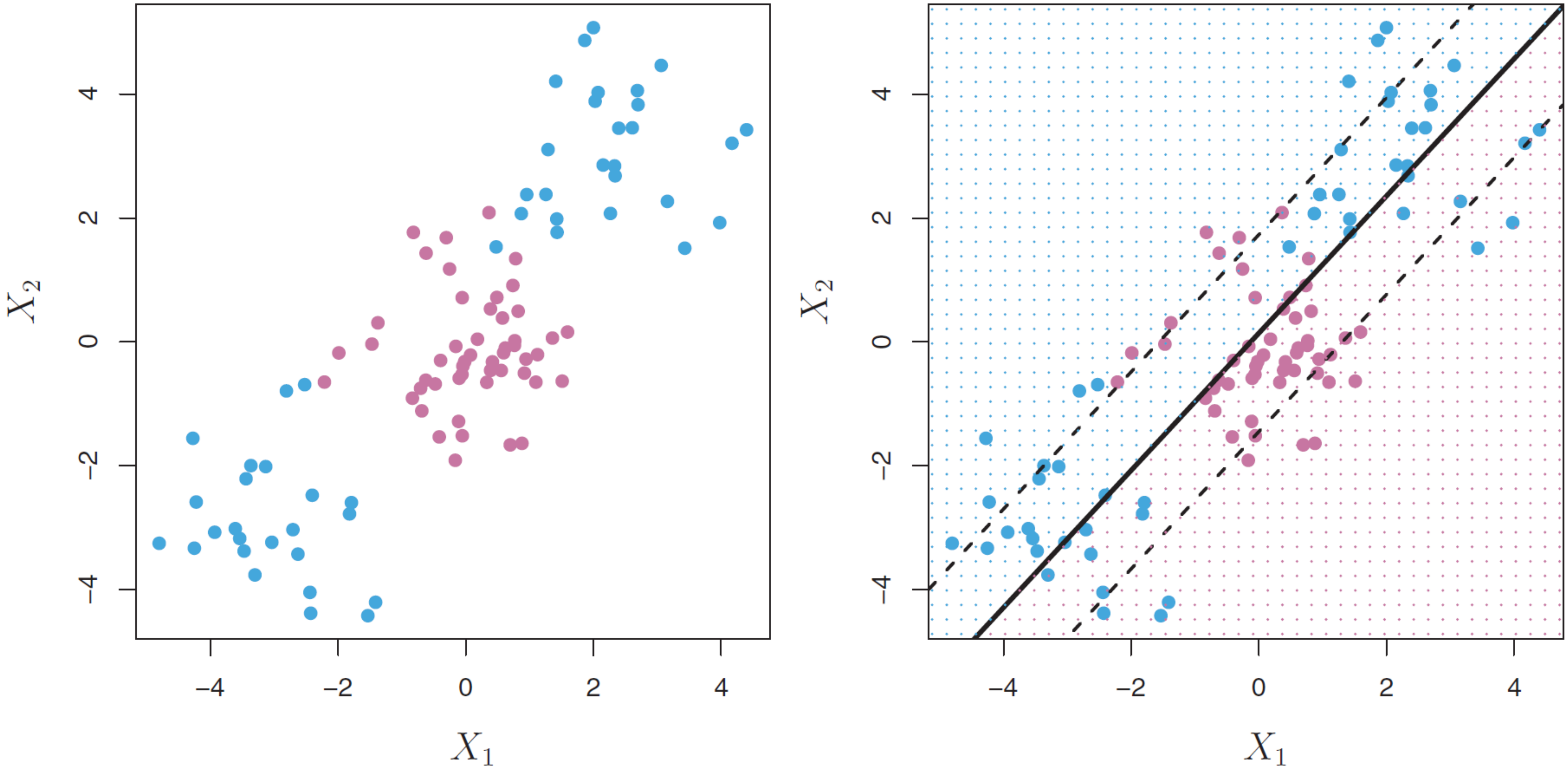
This is sometimes called a soft margin classifier, because errors are allowed.
No more than ‘C’ observations can be on the wrong side of the decision boundary.

Effect of ‘C’ on the Support Vector Classifier

- Larger values of ‘C’ yield …
  - larger margins; more support vectors
  - lower variance; higher bias
- Smaller values of ‘C’ yield …
  - smaller margins; less support vectors
  - lower bias; higher variance
- The margins (dashed lines) are the values for which the absolute value of the decision value is *M*
- The slack value is positive when $y_i\left(\beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_p x_{i,p}\right) < M$



Classification Problem:
Not Linearly Separable (linear model bad)



We could explicitly fit a Support Vector Classifier with Polynomial Predictors …

$$\underset{\beta_0,\beta_{11},\beta_{12}\ldots,\beta_{p1},\beta_{p2},\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M$$

$$\text{subject to } y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_{j1} x_{ij} + \sum_{j=1}^{p} \beta_{j2} x_{ij}^2 \right) \geq M(1 - \epsilon_i),$$

$$\sum_{i=1}^{n} \epsilon_i \leq C, \quad \epsilon_i \geq 0, \quad \sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1.$$

… but the prevalent strategy is to use kernel functions!

The Support Vector Machine

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \boldsymbol{e}^T \boldsymbol{\alpha}$$

$$\text{subject to} \quad \boldsymbol{y}^T \boldsymbol{\alpha} = 0$$

$$0 \leq \alpha_i \leq \textbf{cost}$$

The *cost* parameter refers to the cost of a margin violation: cost is inversely related to the ‘C’ we talked about earlier; but cost is the parameter used by the svm() function.

$$Q_{ij} \equiv y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$\boldsymbol{e} = [1, \ldots, 1]^T \text{ is the vector of all ones}$$

Kernel Functions

- A kernel function measures the similarity between two vectors
- Popular choices include …
  - The "Linear" (dot product) kernel $$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}$$
  - The "Polynomial" kernel $$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d$$
    'd' degree parameter
    [larger degree, higher complexity]
  - The "Gaussian" (radial basis function) kernel $$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$$
    'gamma' bandwidth parameter
    [larger gamma, smaller bandwidth]

- So our new decision function is … $$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i)$$

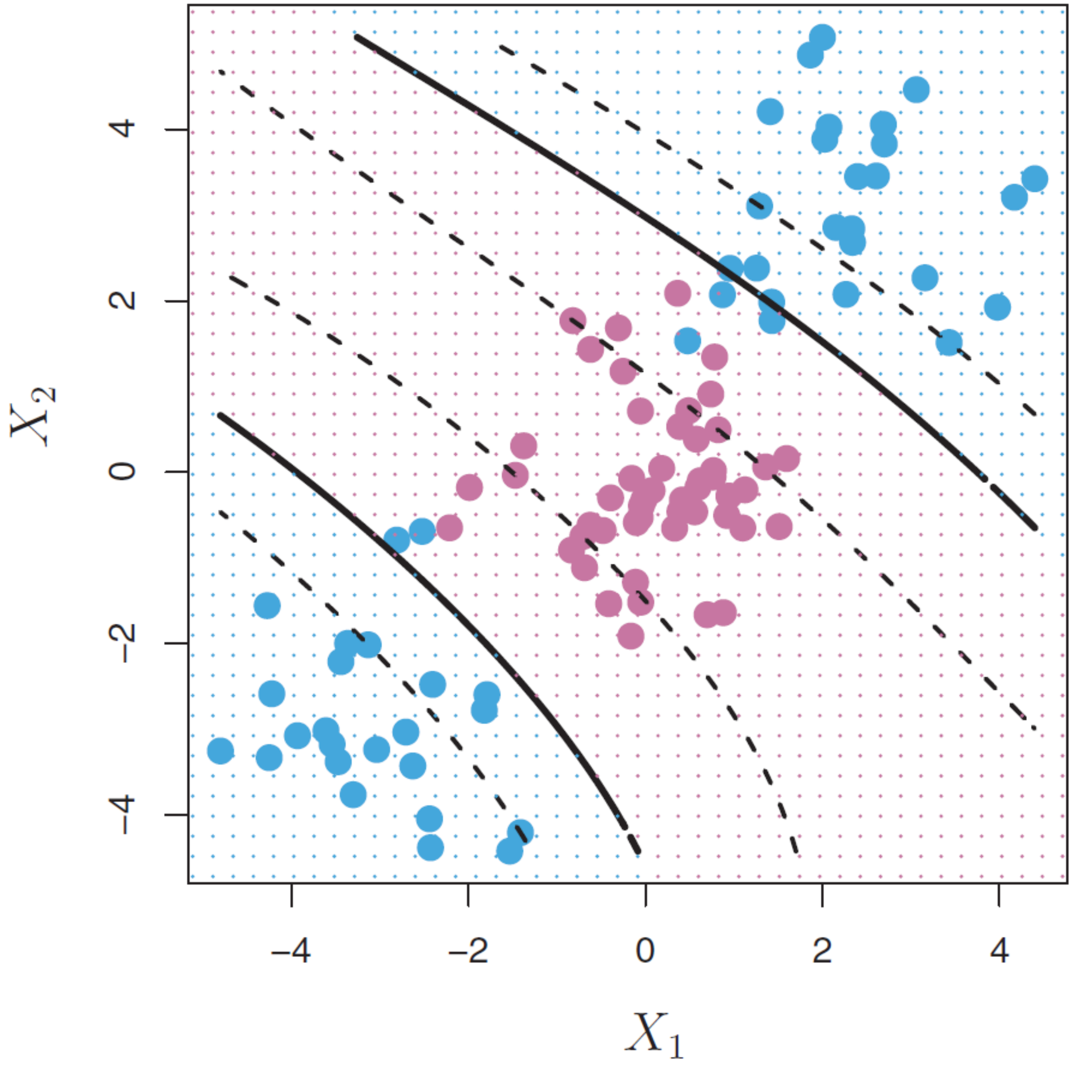
Kernel Functions Viewed as $\langle \phi(x_i), \phi(x_{i'}) \rangle$

- A non-linear kernel function can be viewed as the dot product of a higher dimensional feature space (with a linear hyperplane)
- Consider a classification problem with only one predictor, where the positive class resides on the interval [-2, 2] and the negative class resides on the intervals (-∞, -2) and (2, ∞)
  - This problem is not linearly separable with the original feature space
  - This problem is linearly separable with the higher dimensional feature space provided by a polynomial kernel with degree = 2 [$x_i^2 \leq 4$ => positive class]

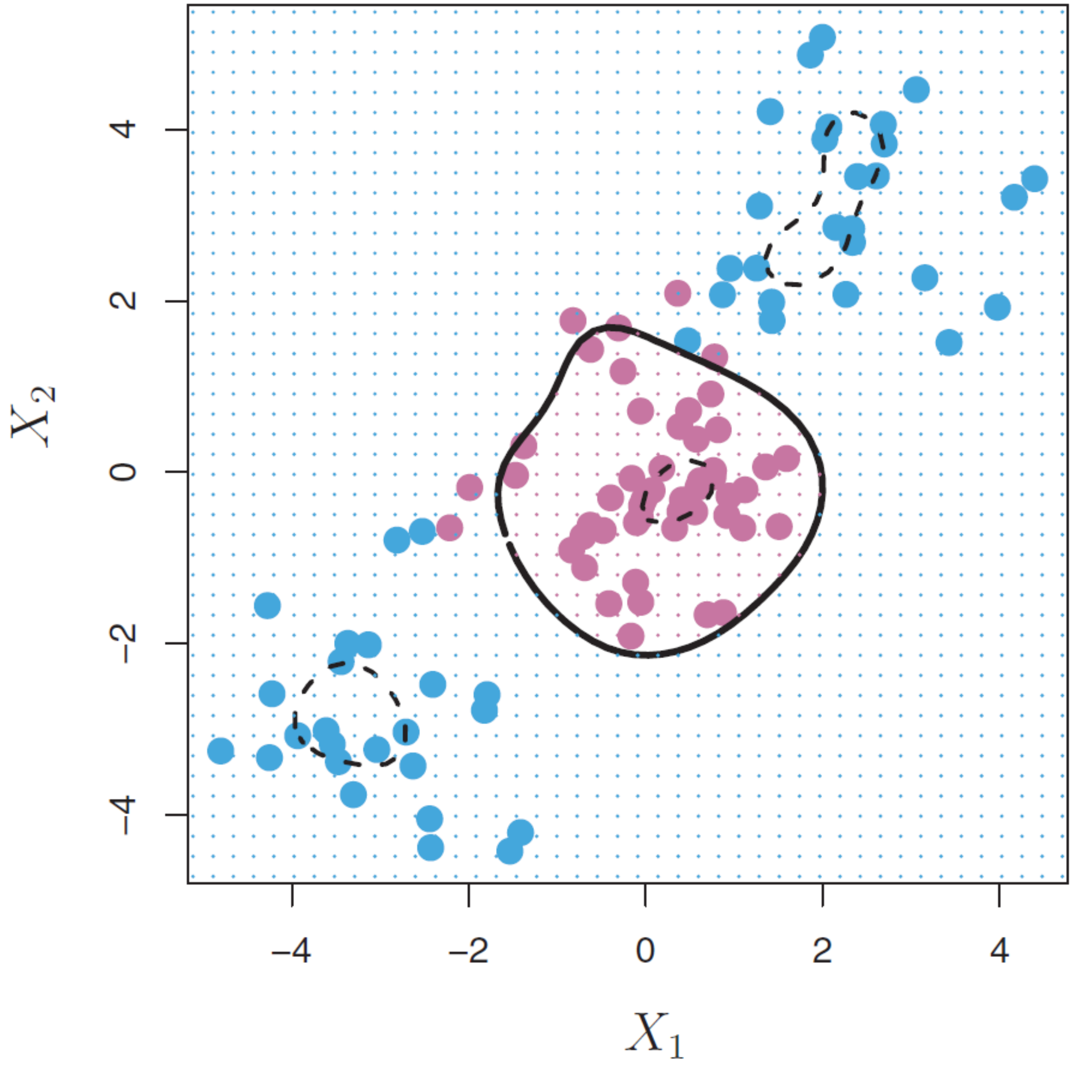
$$K(x_i, x_{i'}) = (1 + x_i x_{i'})^2 = 1 + 2x_i x_{i'} + x_i^2 x_{i'}^2 = \begin{bmatrix} 1 & \sqrt{2}x_i & x_i^2 \end{bmatrix} \begin{bmatrix} 1 \\ \sqrt{2}x_{i'} \\ x_{i'}^2 \end{bmatrix} = \phi(x_i)^T \phi(x_{i'})$$

For simplicity, we'll simply view the number of support vectors as the number of features for our model

Non-Linear Classifiers via Non-Linear Kernel

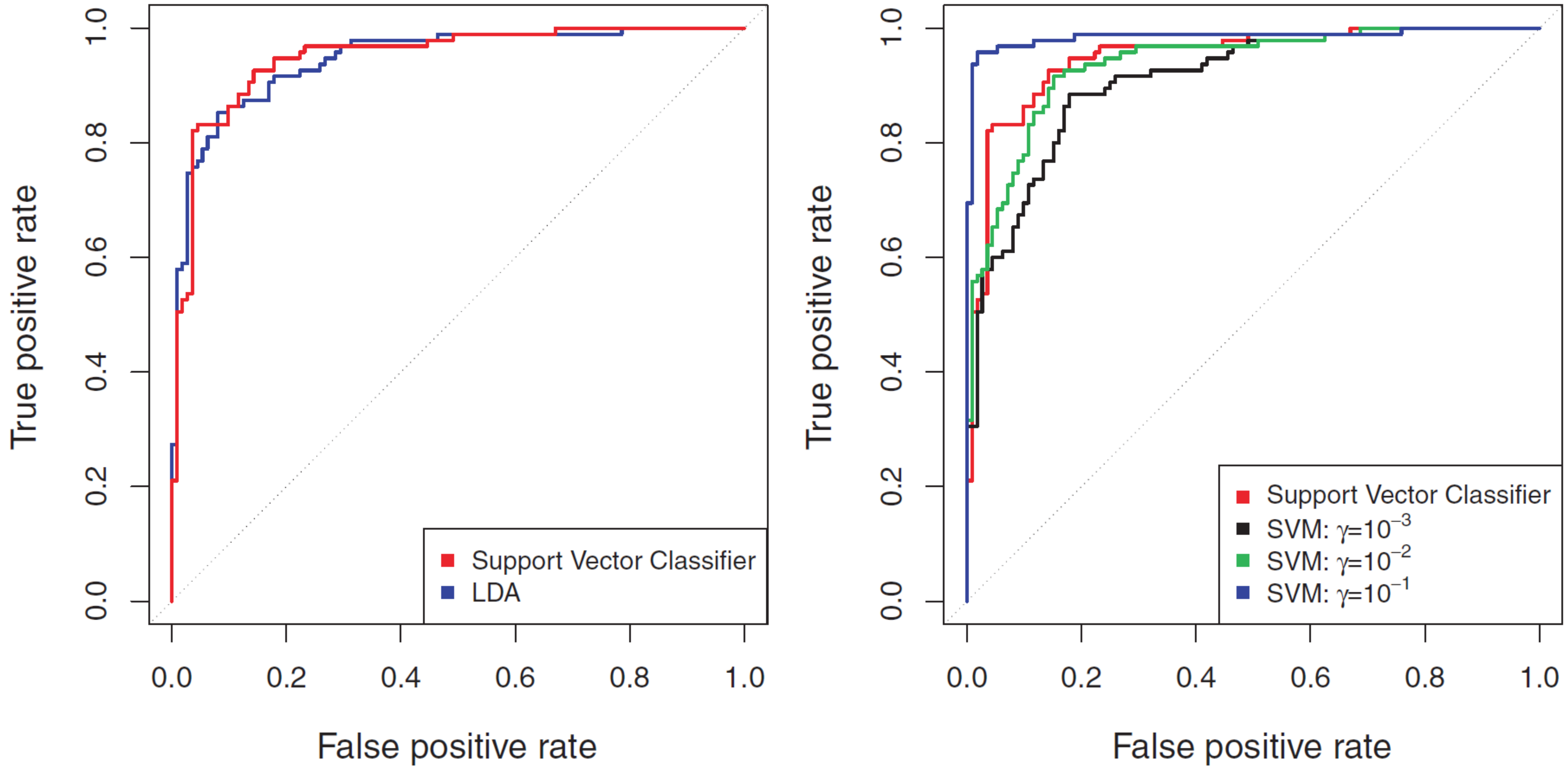


Polynomial Kernel

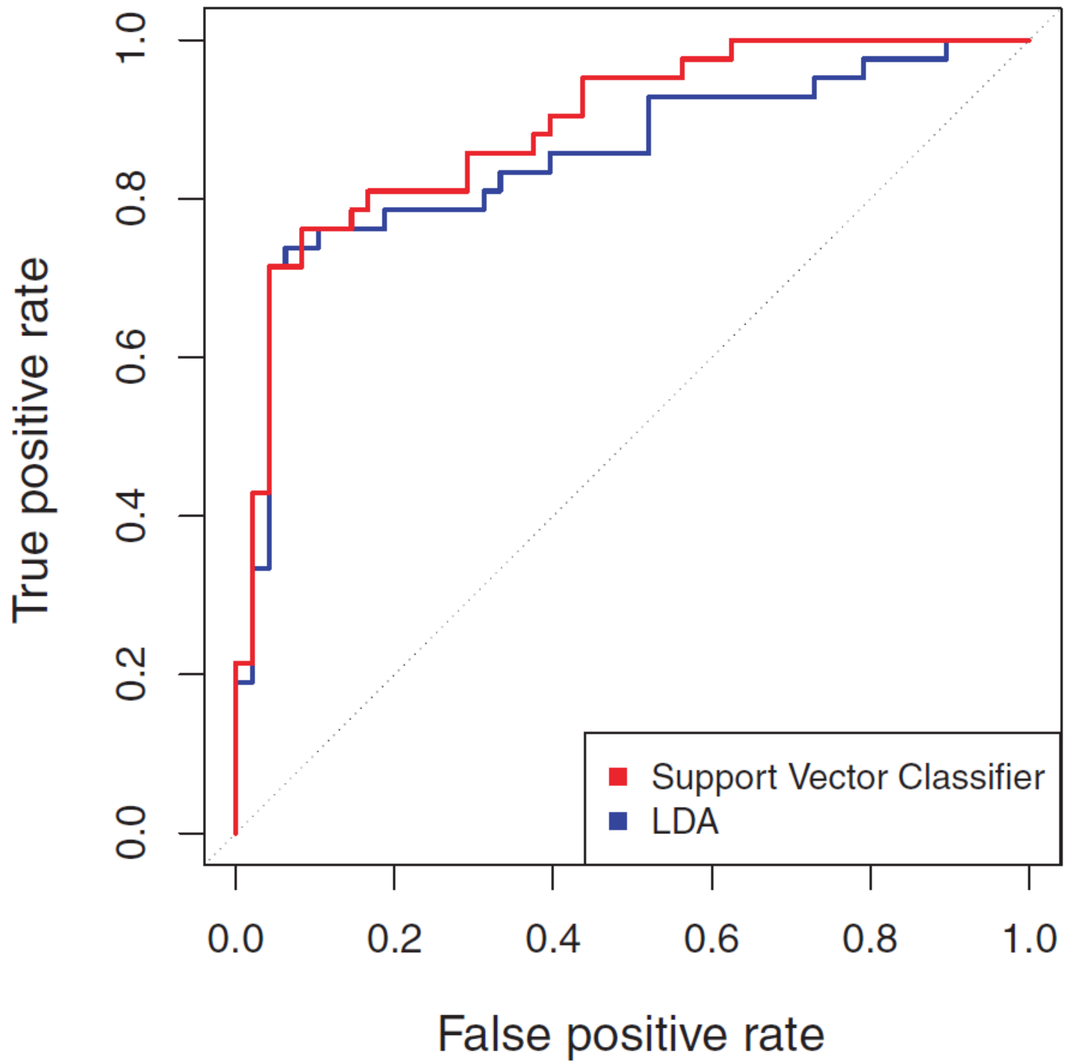


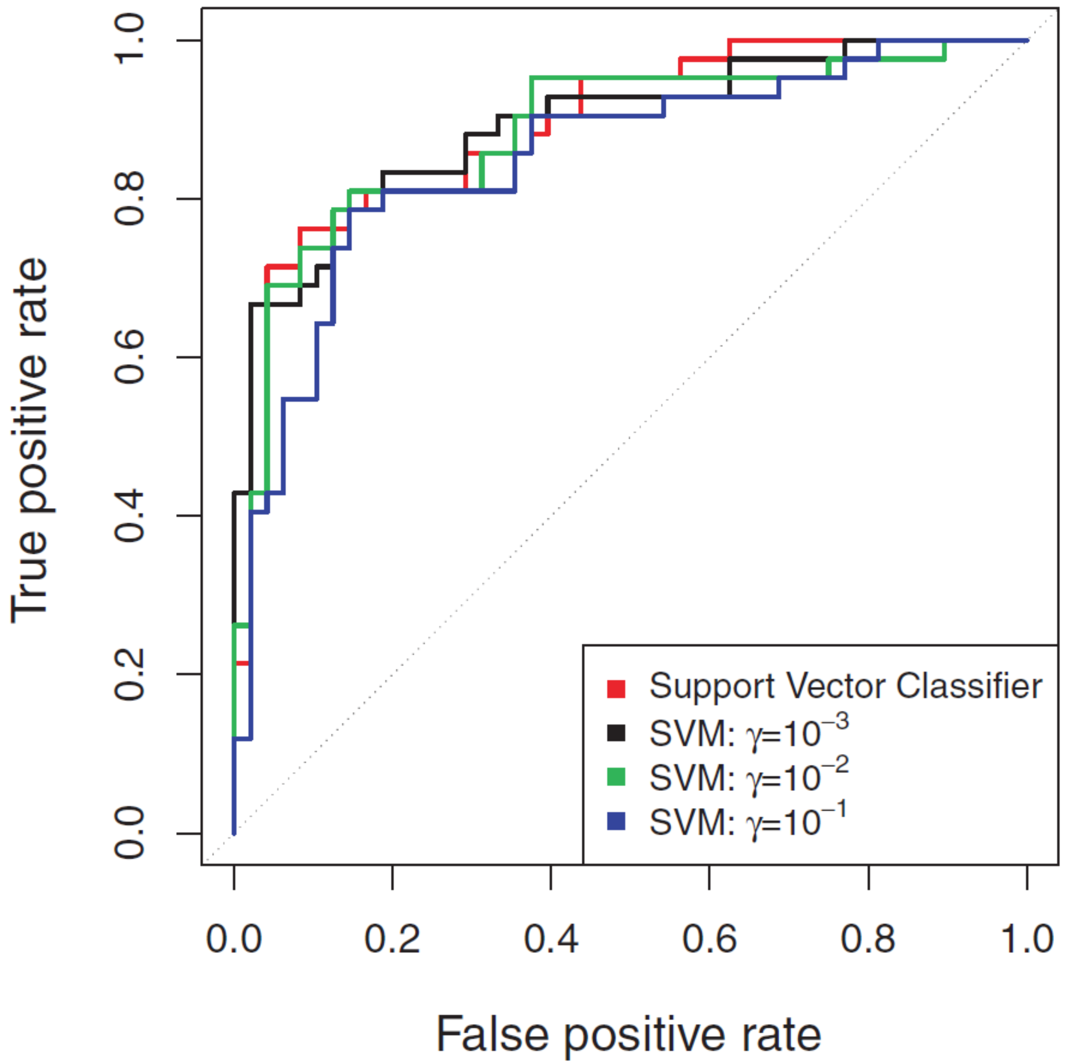
Radial Kernel

Results on the Heart Training Data ☹



Results on the Heart Testing Data





SVMs with ‘K’ (More than Two) Classes

- One-Versus-One Classification
  - Construct $\binom{K}{2}$ classifiers, then assign a test observation to the most frequent class
- One-Versus-All Classification
  - Construct K classifiers, then assign a test observation to the class with the largest response

Relationship to Logistic Regression

We can rewrite this …

$$\underset{\beta_0,\beta_{11},\beta_{12}\ldots,\beta_{p1},\beta_{p2},\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M$$

$$\text{subject to } y_i\left(\beta_0 + \sum_{j=1}^{p}\beta_{j1}x_{ij} + \sum_{j=1}^{p}\beta_{j2}x_{ij}^2\right) \geq M(1-\epsilon_i),$$

$$\sum_{i=1}^{n}\epsilon_i \leq C, \quad \epsilon_i \geq 0, \quad \sum_{j=1}^{p}\sum_{k=1}^{2}\beta_{jk}^2 = 1.$$

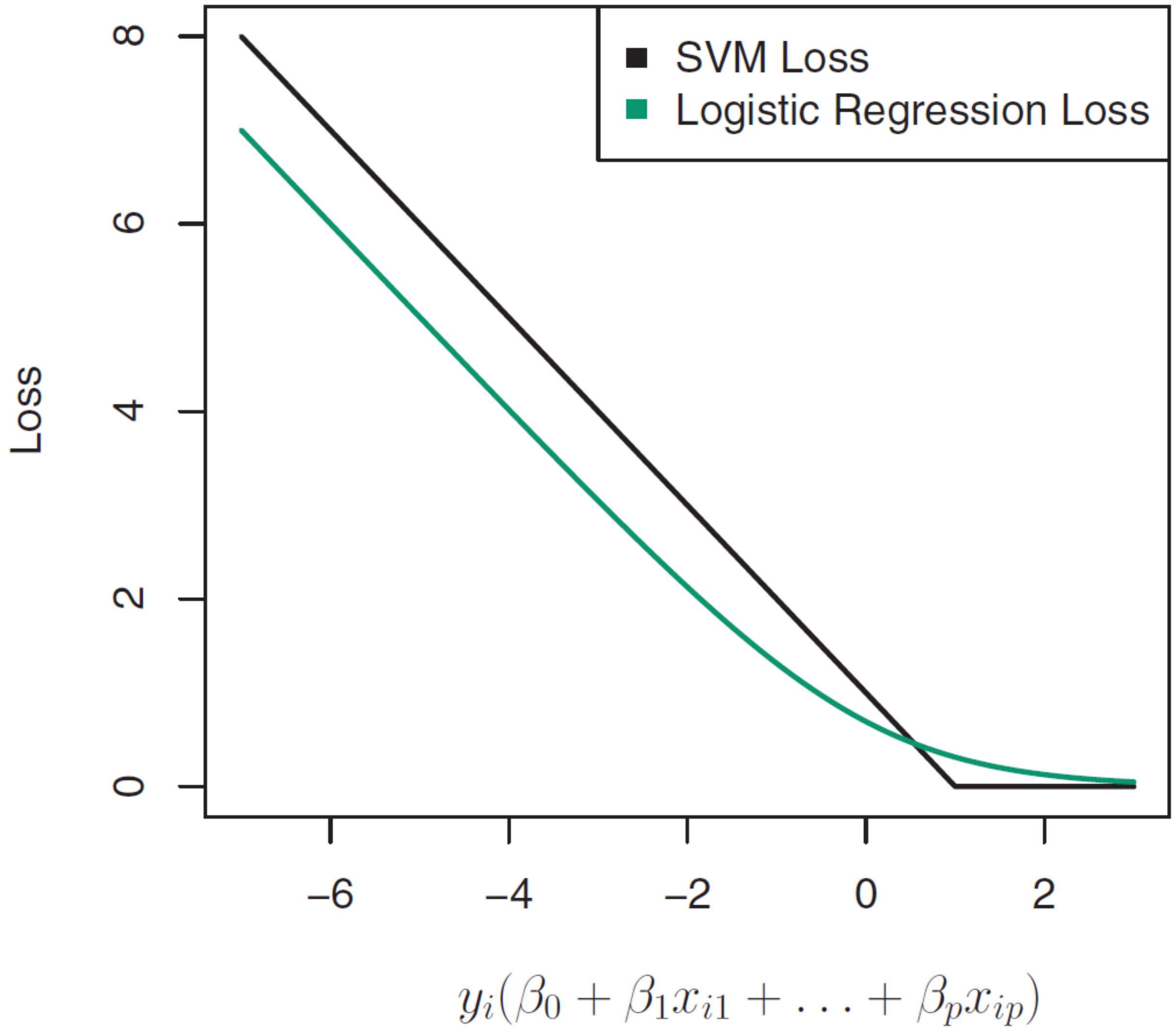
… as this …

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{minimize}}\left\{\sum_{i=1}^{n}\max\left[0, 1 - y_i f(x_i)\right] + \lambda\sum_{j=1}^{p}\beta_j^2\right\}$$

… which has this familiar form …

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{minimize}}\left\{L(\mathbf{X},\mathbf{y},\beta) + \lambda P(\beta)\right\}$$

SVM (Hinge) Loss versus Logistic Regression (Log) Loss



Agenda