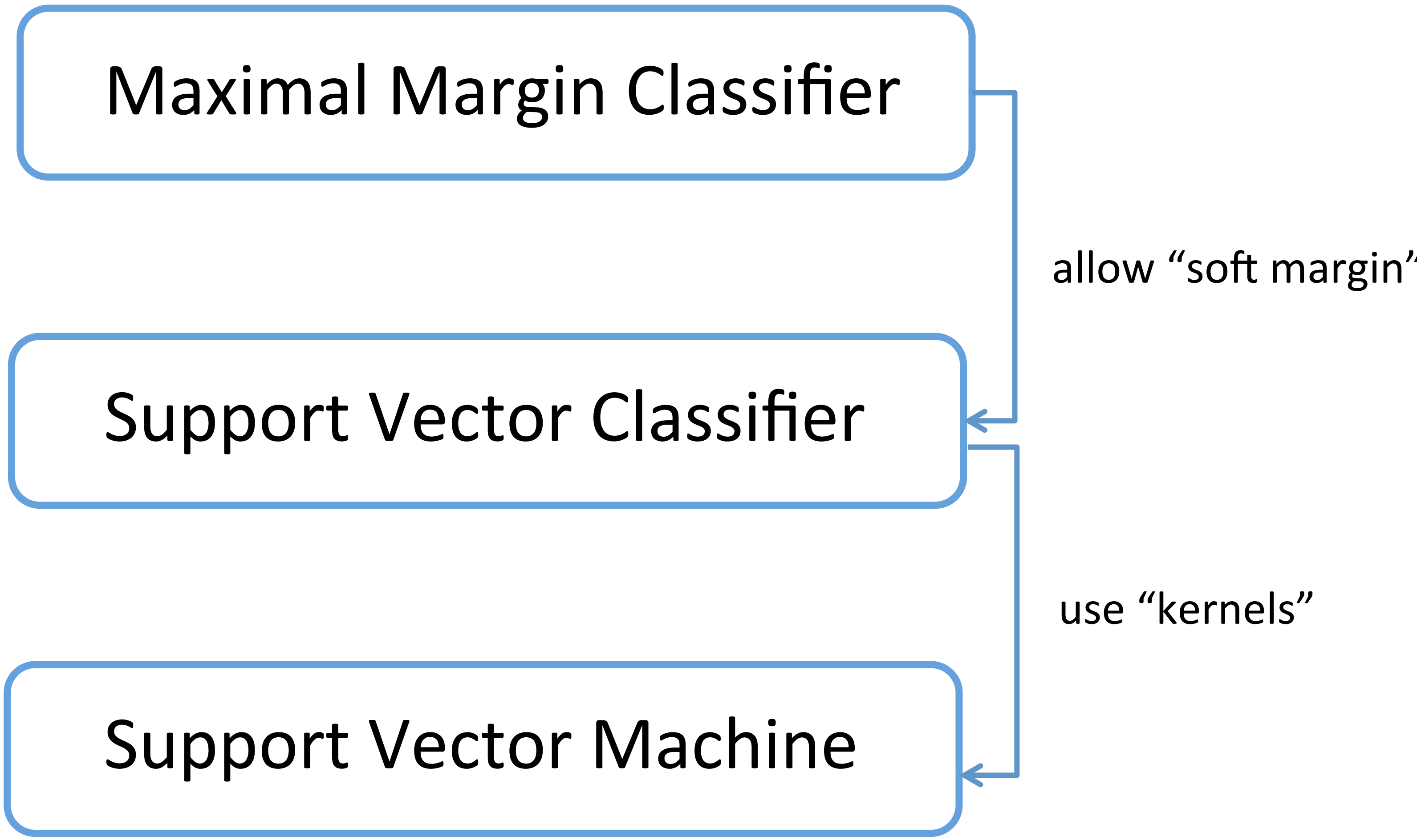
Support Vector Machines

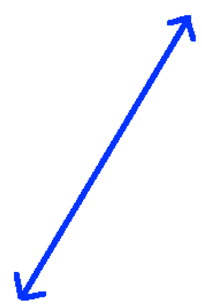
"Some figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani "


Maximal Margin Classifier
allow “soft margin”
Support Vector Classifier
use “kernels”
Support Vector Machine

Hyperplanes

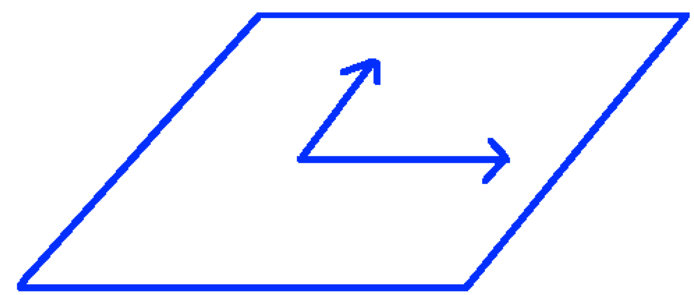
In p-dimensional space, a flat affine subspace of dimension p-1

p = 2



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

p = 3

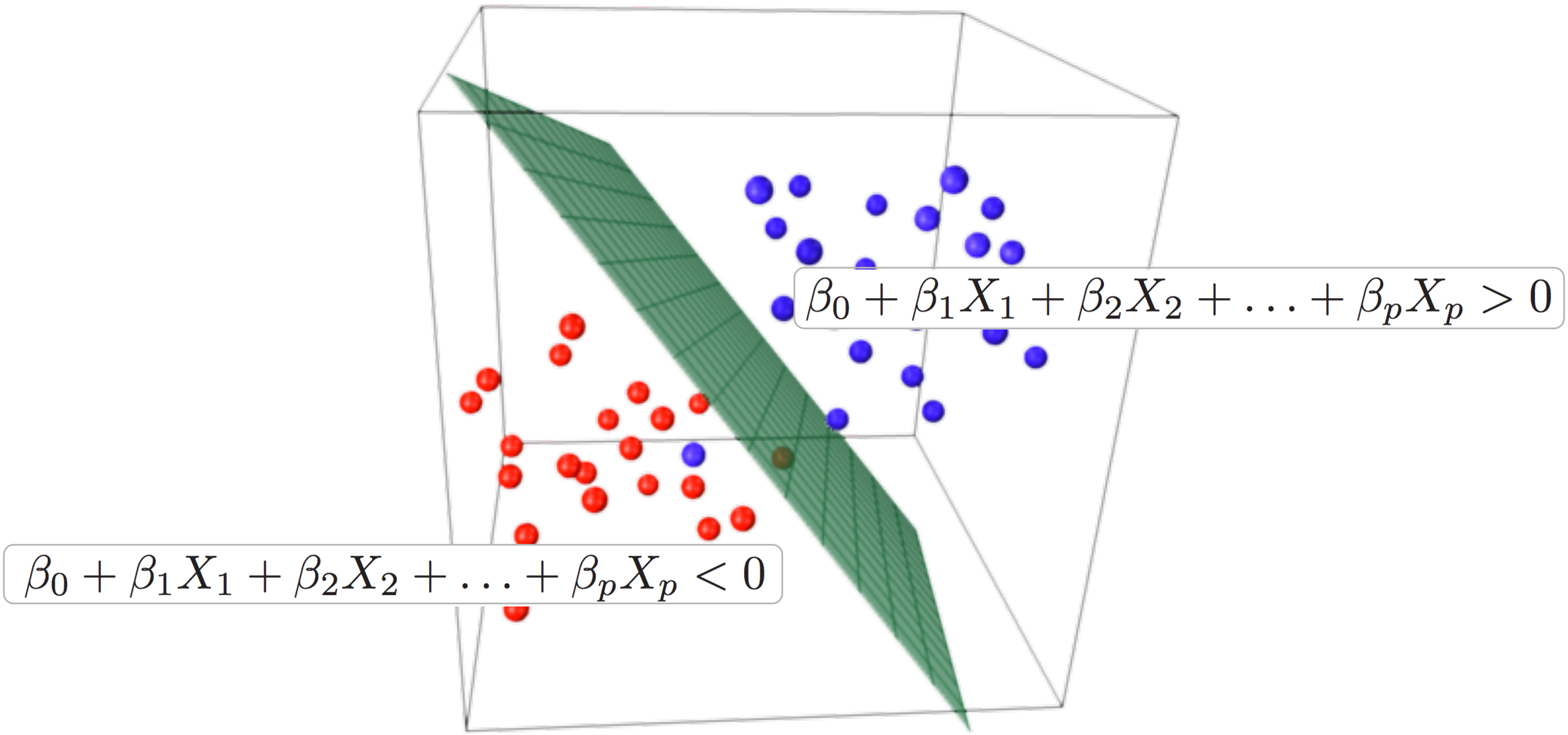


$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 = 0$$

Generally, a hyperplane in p-dimensional space can be defined as

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

Hyperplanes



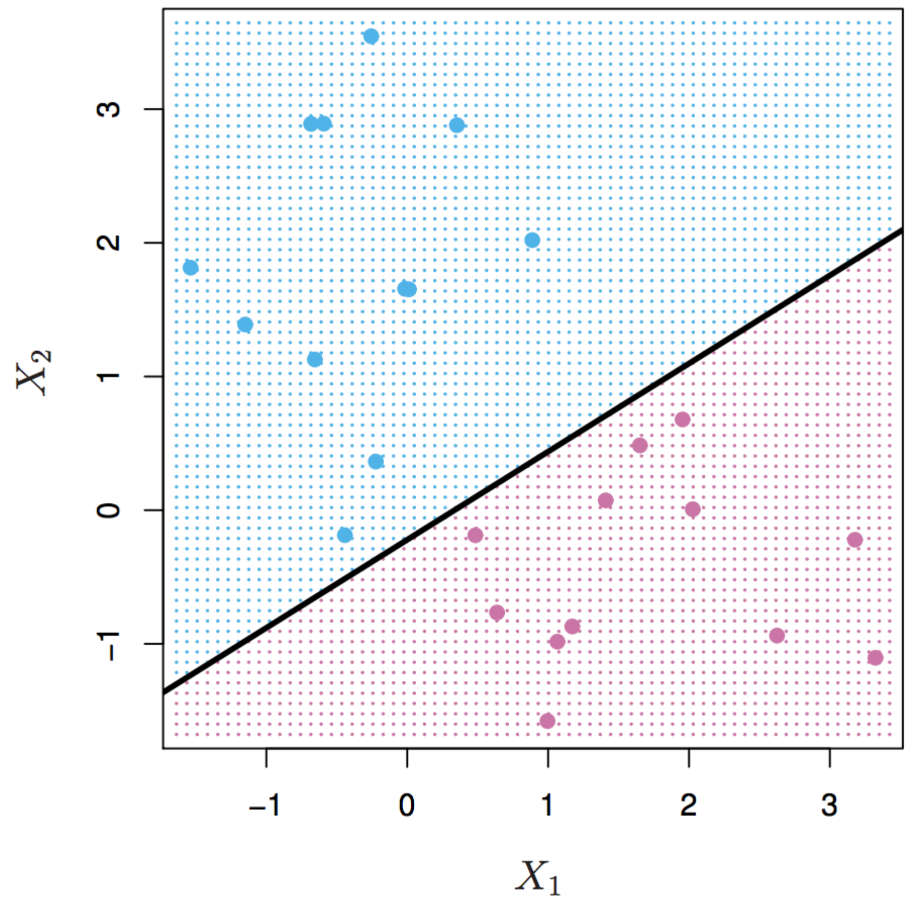
Can think of hyperplane as dividing p-dimensional space into two halves

Separating Hyperplane

Suppose we code…

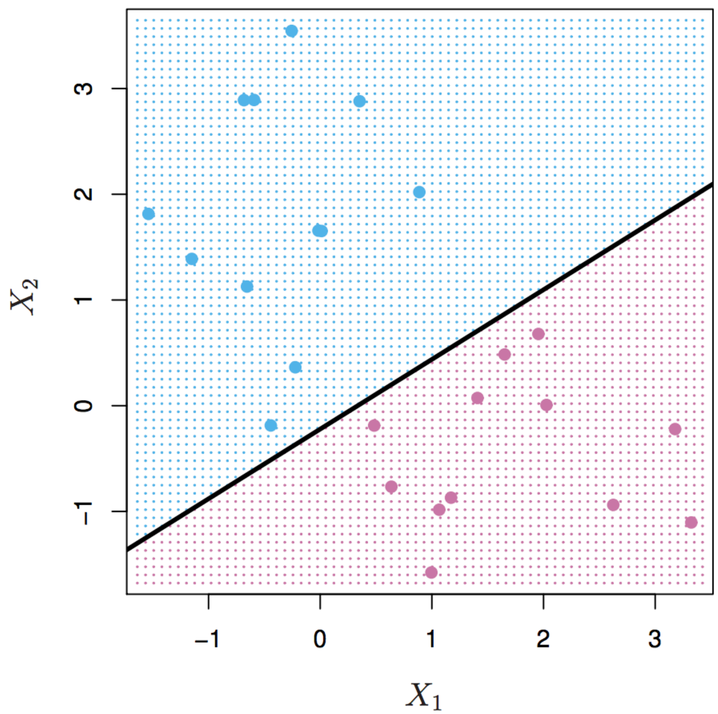
If y_i = Blue $\Longrightarrow$ y_i = +1

If y_i = Red $\Longrightarrow$ y_i = -1



We have a *separating hyperplane*,
if for *all points*, we have...

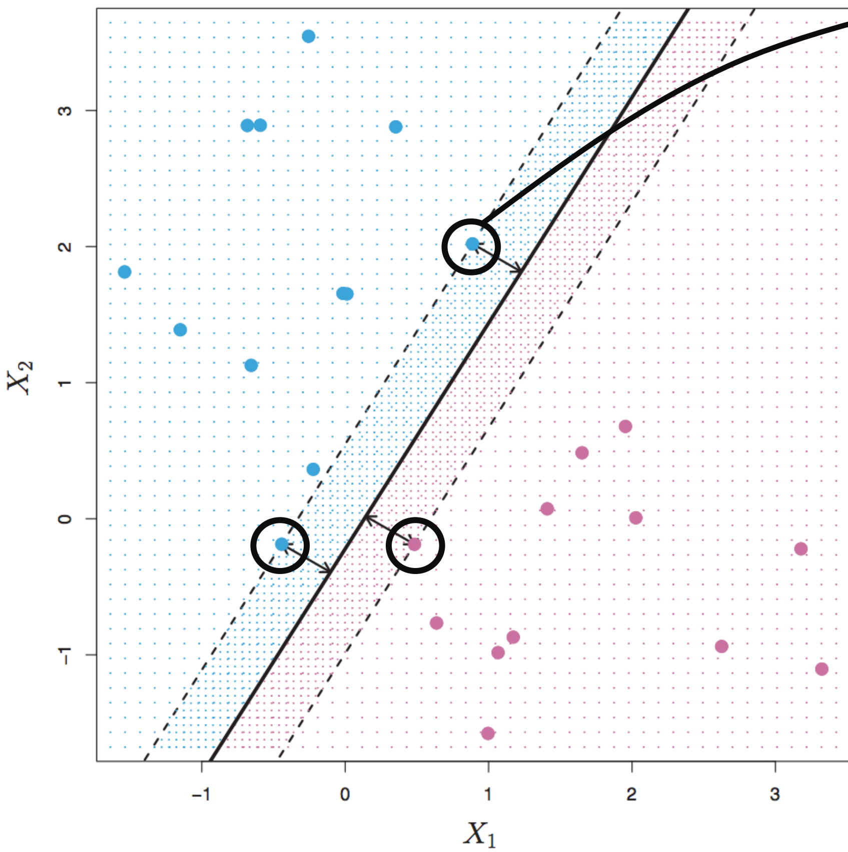
$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} > 0 \;\; when \;\; y_i = +1$$



$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} < 0 \;\; when \;\; y_i = -1$$

In particular, we fit…

“Support Vectors”

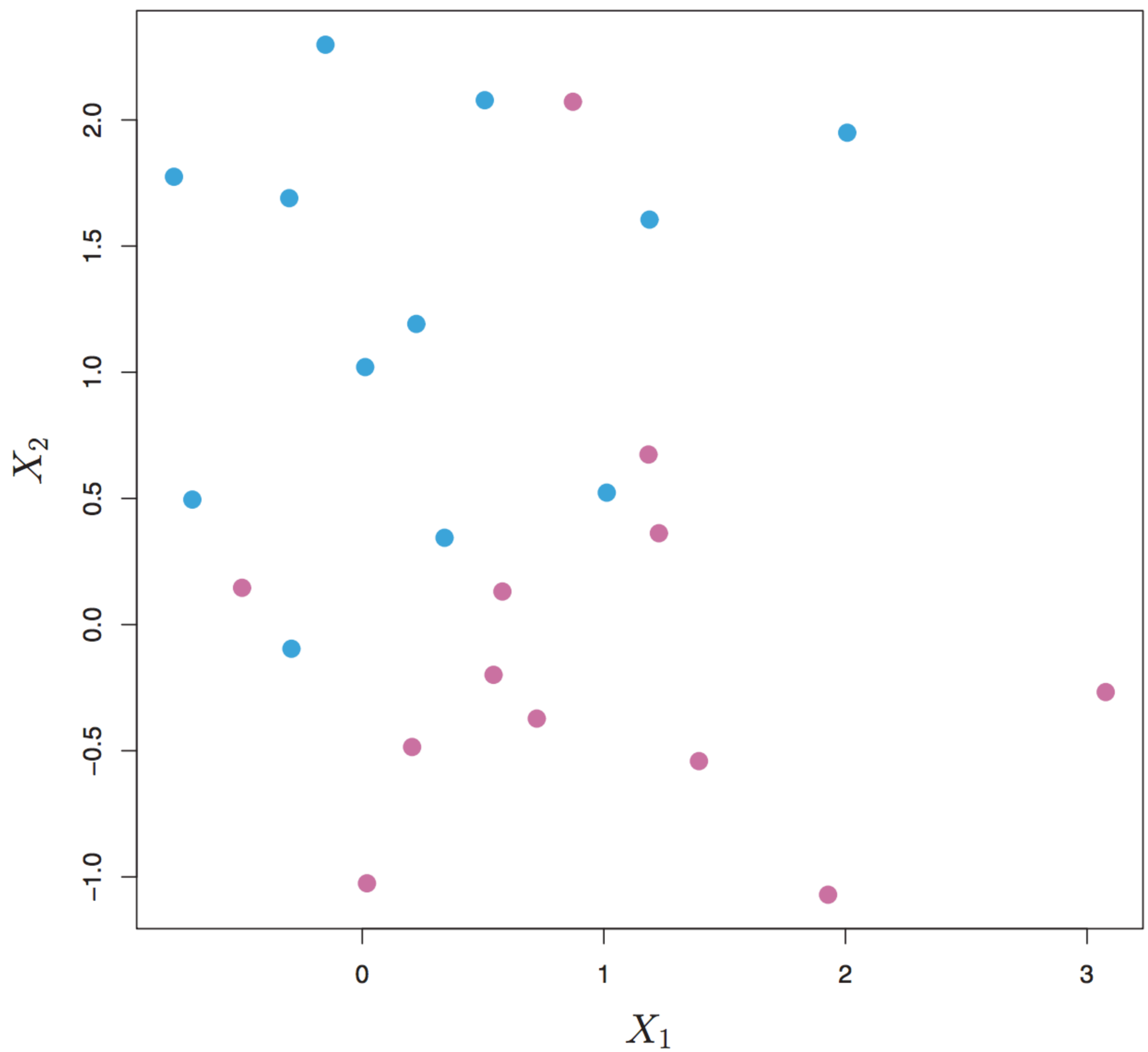


$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\, M$$

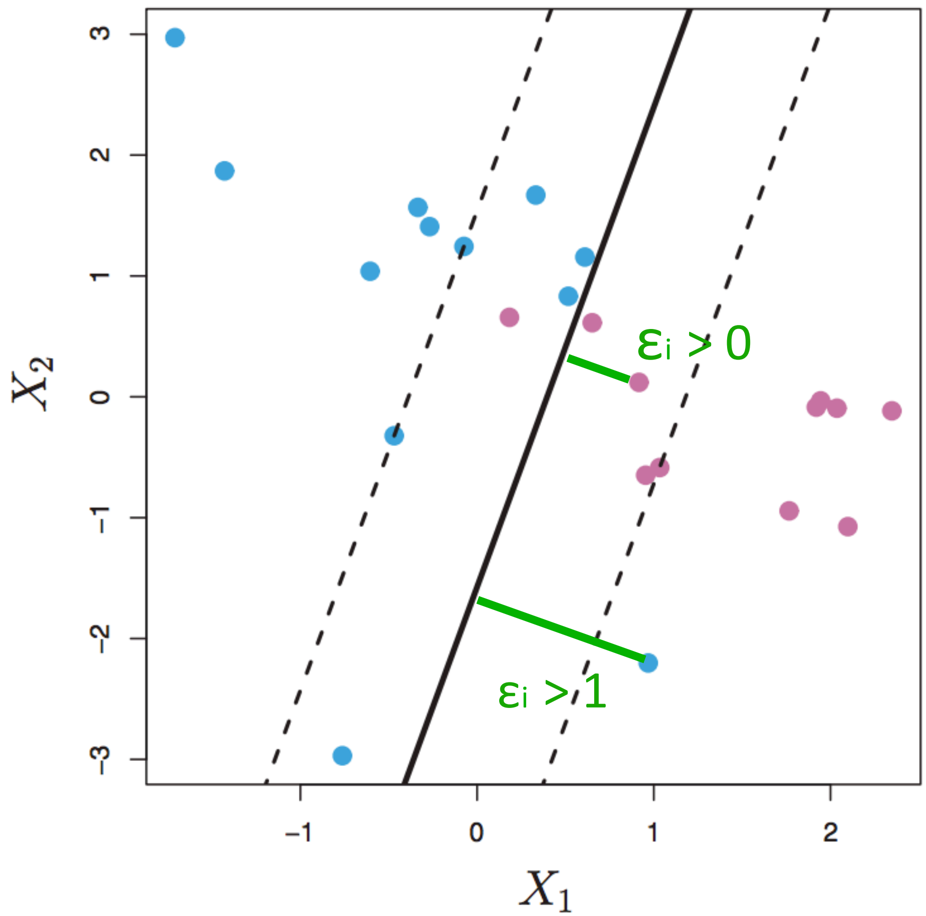
$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

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

hmm....



need some sort of *budget*



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

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

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

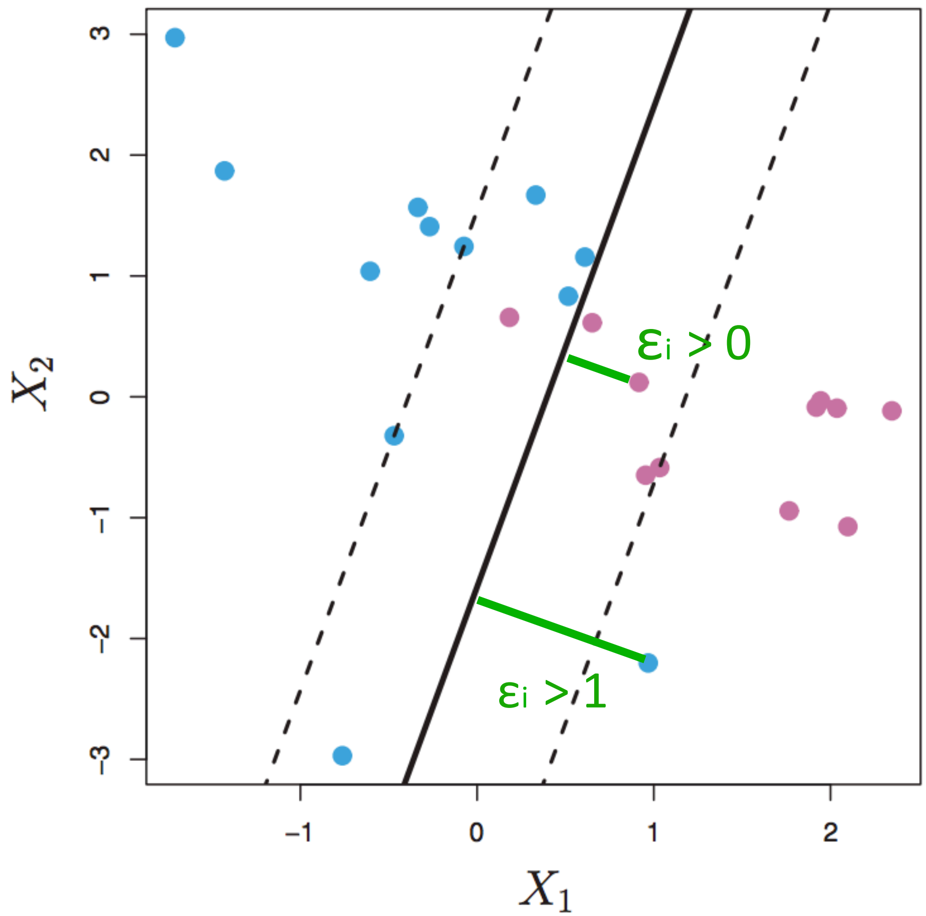
$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,$$

Slack from each point

Budget that we can tune

εi = 0 for being on correct side of margin
εi > 0 for violating the margin
εi > 1 for being on wrong side of hyperplane

need some sort of *budget*



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

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

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

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

Slack from each point

Budget that we can tune

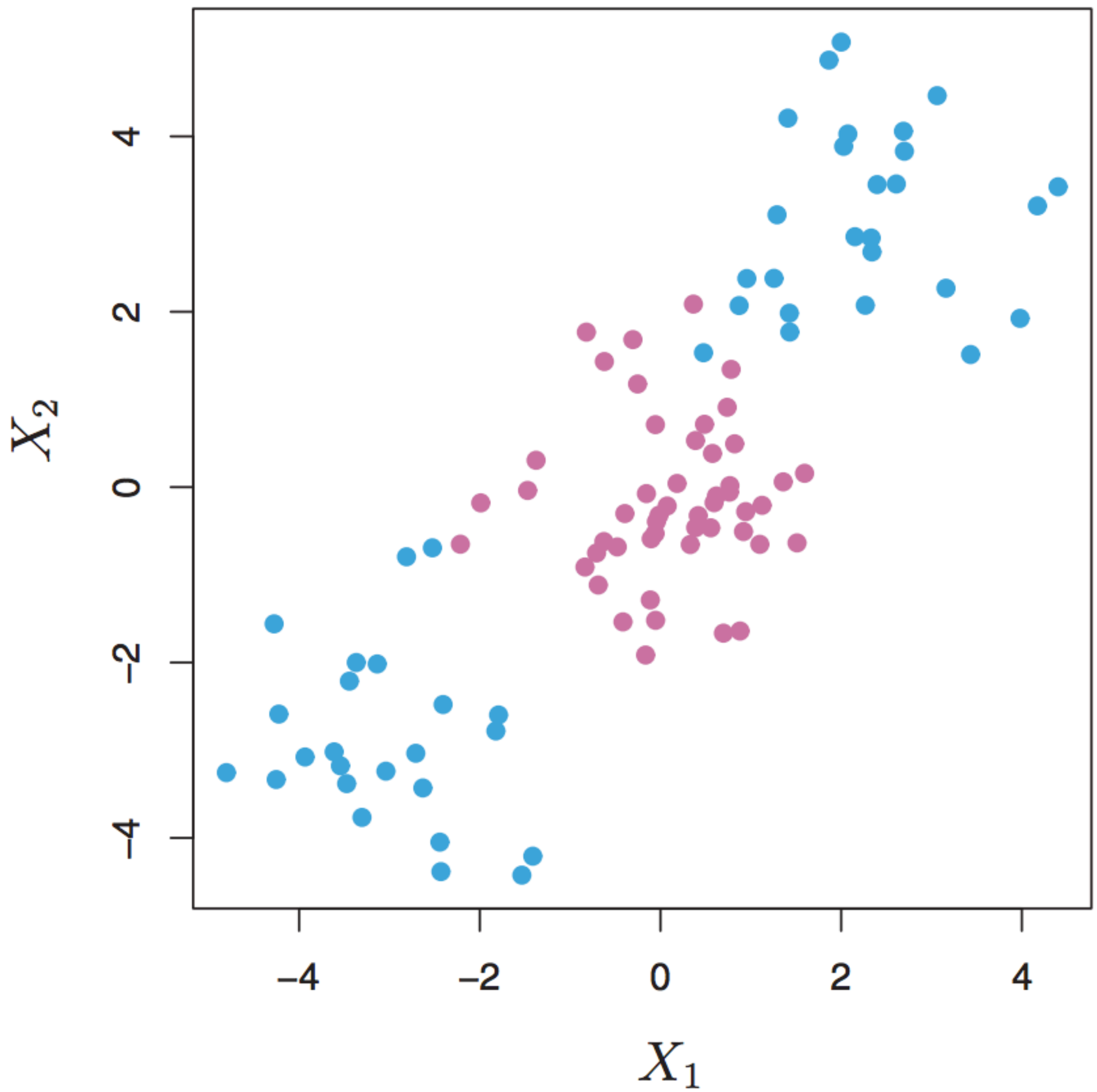
Bias Variance Tradeoff

- C small ⬄ Low bias, High Variance
- C large ⬄ High bias, Low Variance
  (not quite as clear cut)

$\varepsilon_i = 0$ for being on correct side of margin
$\varepsilon_i > 0$ for violating the margin
$\varepsilon_i > 1$ for being on wrong side of hyperplane

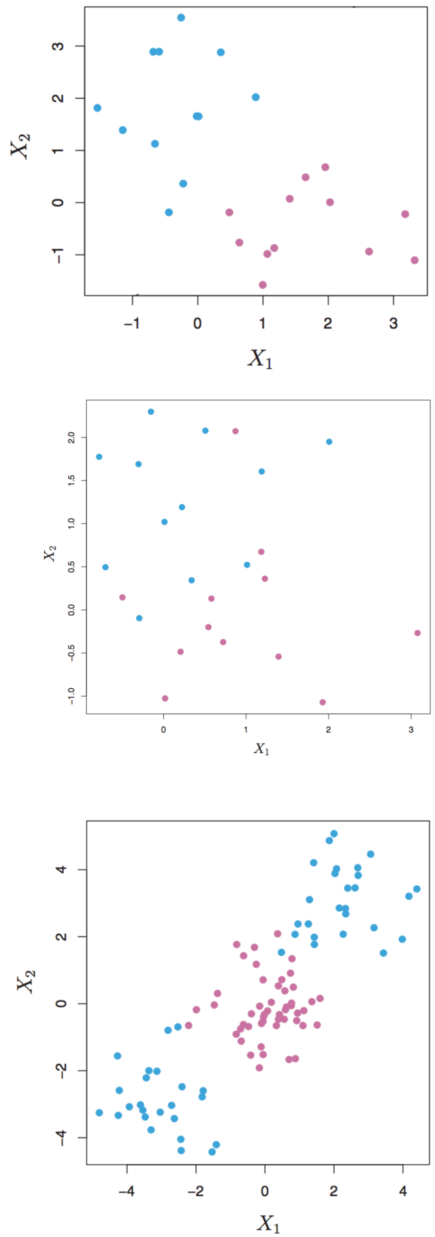
Afternoon

hmm….



[Cool video](https://www.youtube.com/watch?v=3liCbRZPrZA).

https://www.youtube.com/watch?v=3liCbRZPrZA



Maximal Margin Classifier

allow "soft margin"

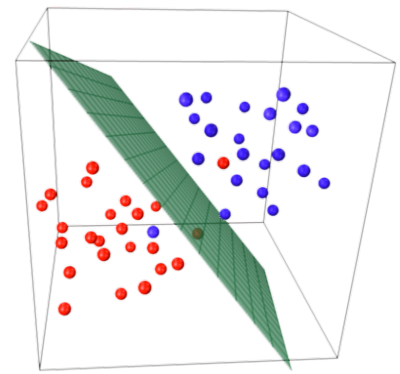
Support Vector Classifier

use "kernels"

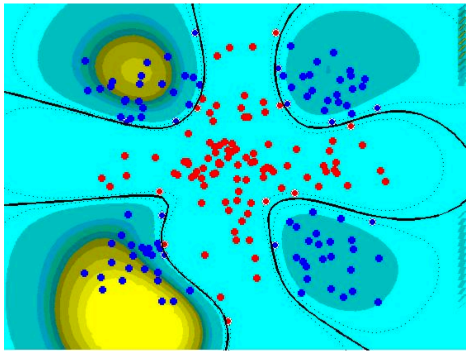
Support Vector Machine

So what are SVMs again?

- Hyperplane that separates data as well as possible, while allowing some room for error ("soft margin")



- Kernels are powerful way to accommodate non-linear class boundaries.



$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i)$$

Kernels

Solution to SVC only involves inner product of observations

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^{p} x_{ij} x_{i'j}$$

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle$$ ← SVC

$$f(x) = \beta_0 + \sum_{\boxed{i \in \mathcal{S}}} \alpha_i \langle x, x_i \rangle$$ ← Only requires support vectors

More generally, instead of just taking inner product, we can use *Kernels*

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i)$$ ← SVM, since using Kernels now

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}$$ Linear Kernel

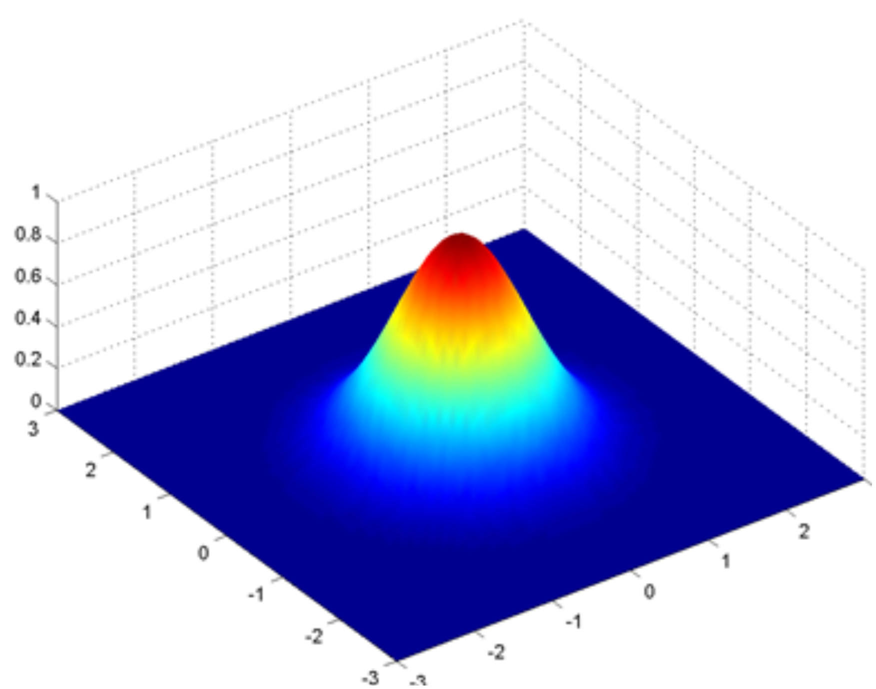
$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d$$ Polynomial Kernel

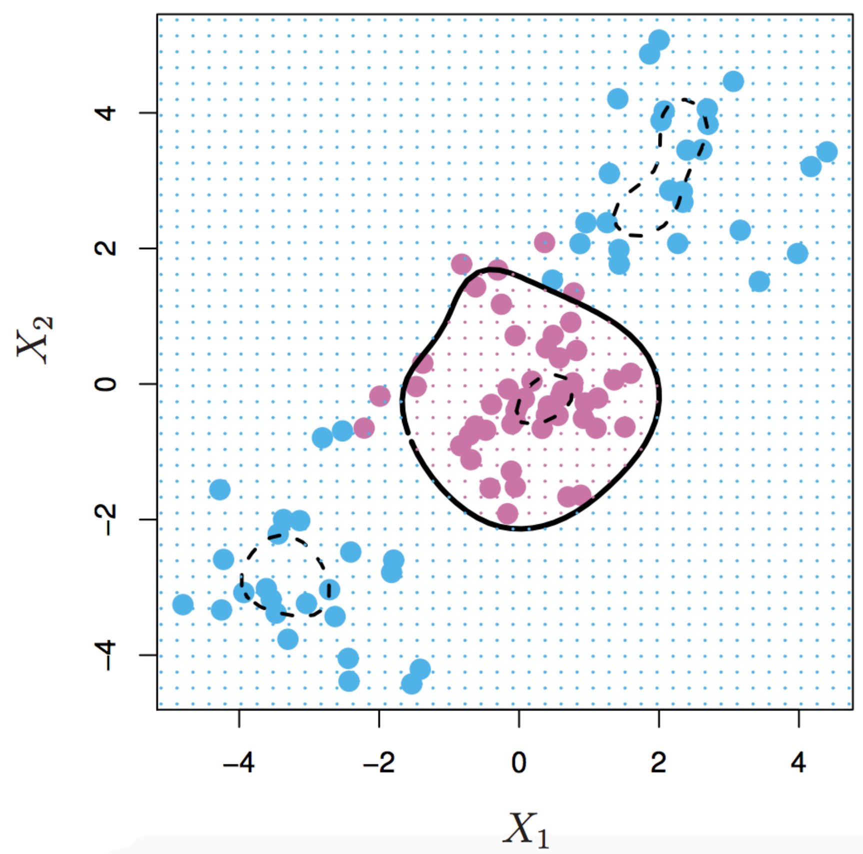
$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$$ Radial Basis Function Kernel ("Gaussian")

Radial Basis Kernel (Gaussian)

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$$

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \hat{\alpha}_i K(x, x_i)$$





Polynomial Kernel

- Expand feature space by simply creating new features

$$(X_1, X_2) \longrightarrow (X_1,\ X_2,\ X_1^2,\ X_2^2,\ X_1X_2)$$

- Same idea of hyperplane decision boundary, but it is non-linear when projected down to X1 vs. X2 space

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2 + \beta_5 X_1 X_2 = 0$$

- Cool Video: https://www.youtube.com/watch?v=3liCbRZPrZA

Turns out, we can rewrite the optimization

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

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

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

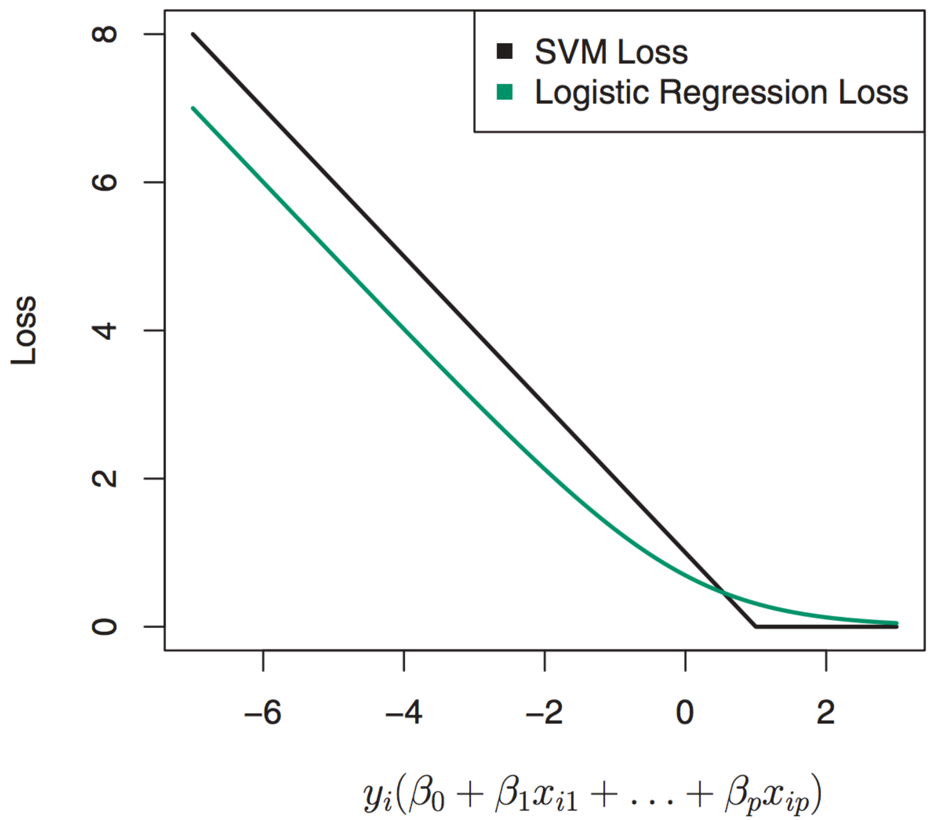
$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,$$

Loss + Penalty

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

Relation to Logistic Regression

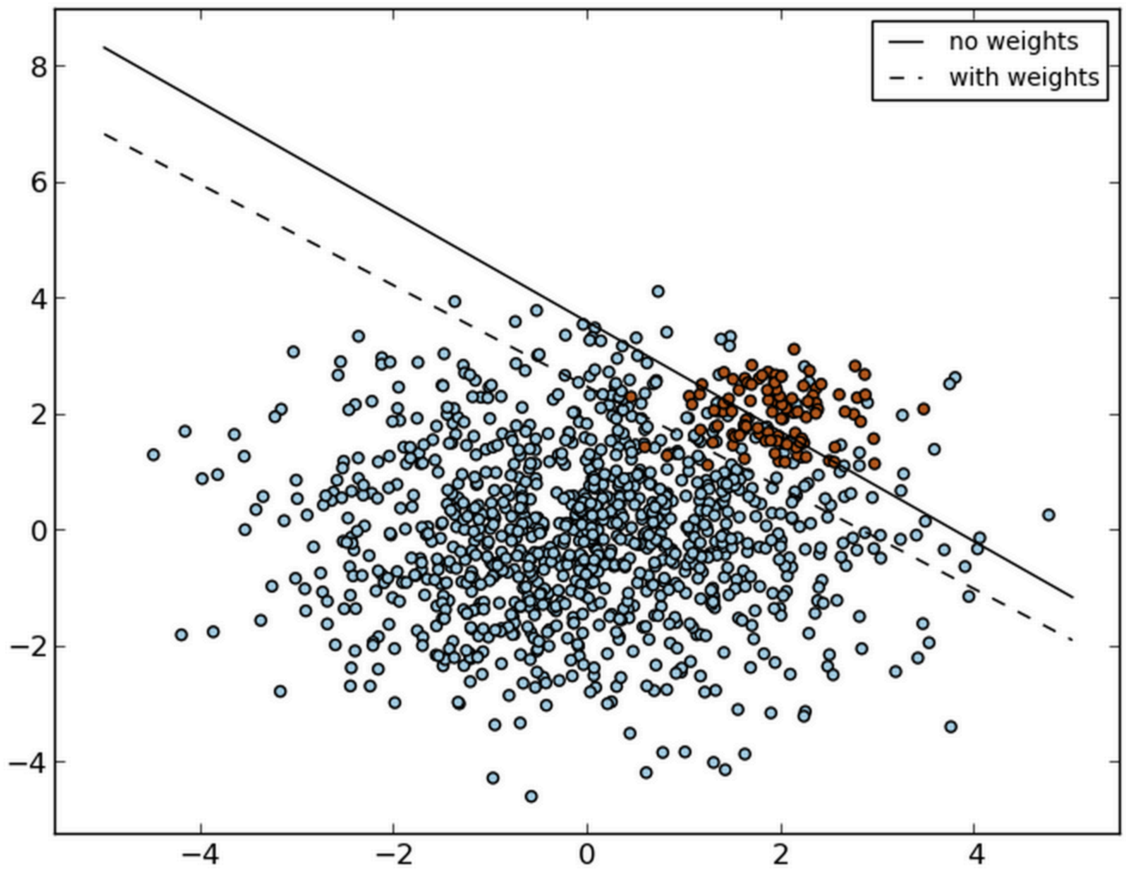
$$\sum_{i=1}^{n} \max\left[0, 1 - y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})\right]$$



- Often results similar to logistic regression

- Robust to observations far from hyperplane

Dealing with Unbalanced Classes



- Suppose 10% Red
- Can adjust weights inversely proportional to class frequencies

Questions

- What's a hyperplane? Specify equation
- How does a Support Vector Classifier work?
  - Describe the "soft margin" aspect
  - Describe the "hinge loss" aspect – Slide 16
  - Contrast with Logistic Regression (as compared to hinge loss)
- How to handle unbalanced classes?
- How to tune?

Appendix

Rewriting the optimization

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

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

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

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

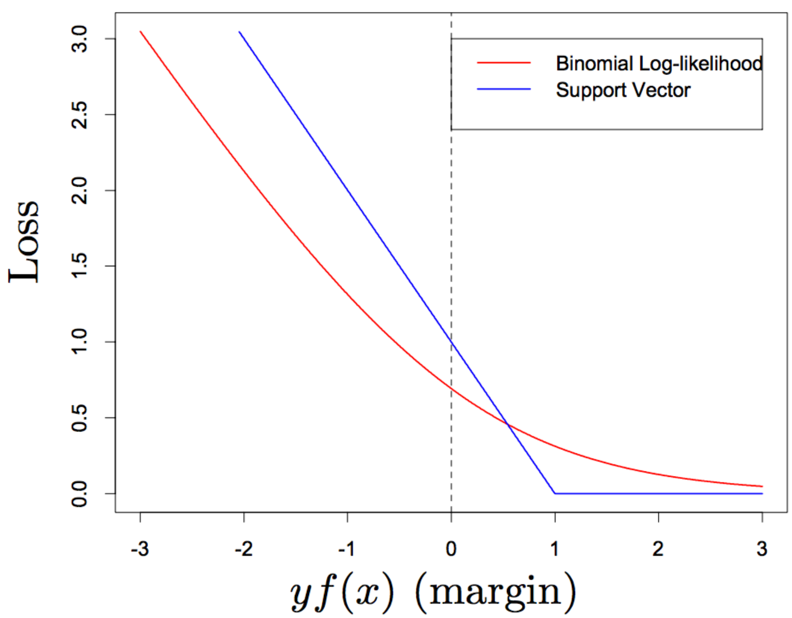
$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{minimize}} \left\{ \sum_{i=1}^{n} \underline{\max\left[0, 1 - y_i f(x_i)\right]} + \underline{\lambda} \sum_{j=1}^{p} \beta_j^2 \right\}$$

The idea is to take a certain budget, C, and find the optimal β vector that achieves the maximum margin possible, M.

- C → λ, both tuning parameters

- ε_i = 0 for being on correct side of margin
  ε_i > 0 for violating the margin
  ε_i > 1 for being on wrong side of hyperplane

More details on SVM vs. Logistic Regression



For Y = -1 or Y = +1

<u>Logistic Regression</u>

- Loss Function, or "binomial Log Likelihood"*

$$L[Y, f(X)] = \log\left(1 + e^{-Yf(X)}\right)$$

- Which estimates the logit

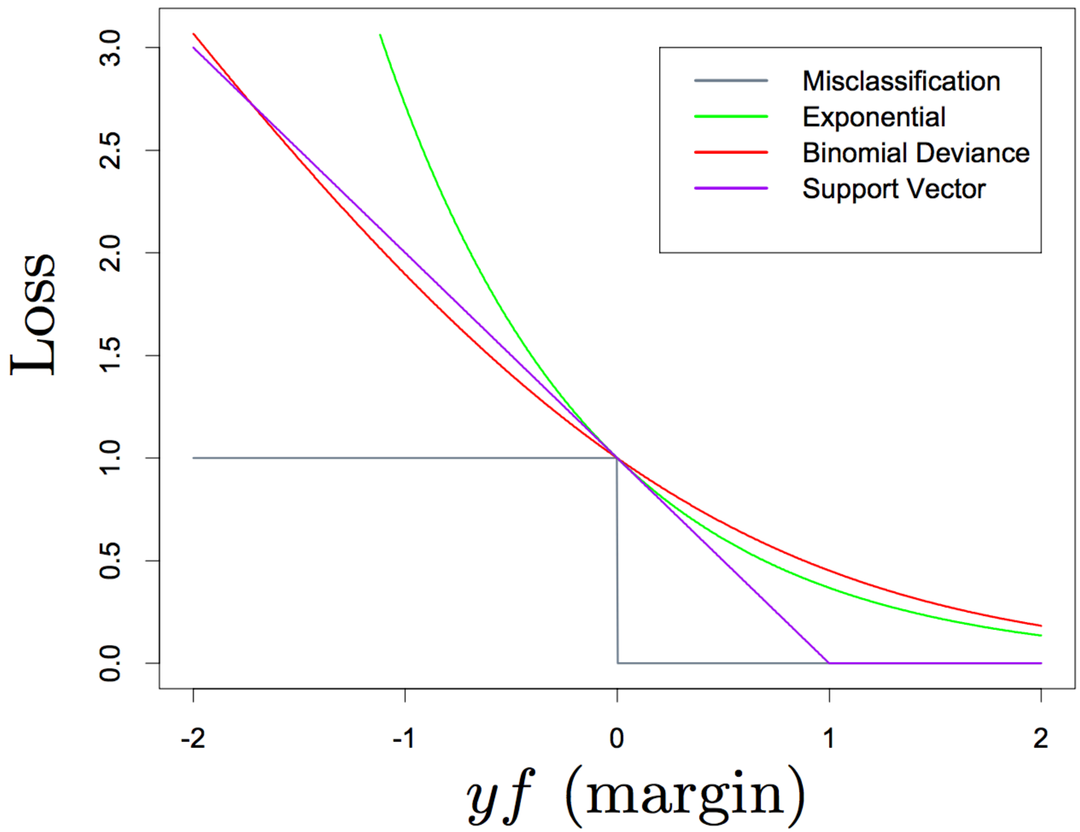
$$f(X) = \log \frac{\Pr(Y = 1|X)}{\Pr(Y = -1|X)}$$

*pg 13: http://www.stat.wisc.edu/~mchung/teaching/MIA/reading/GLM.logistic.Rpackage.pdf

SVMs vs. Logistic Regression

- When classes nearly separable, SVM tends to do better than Logistic Regression
- Otherwise, Logistic Regression (with Ridge) and SVMs are similar
- However, if you want to estimate probabilities, Logistic Regression is the choice.
- With kernels, SVMs work well. Logistic Regression works fine with kernels but can get computationally too expensive

SVM and other Loss functions



- Can consider training model using different loss functions and scoring each one
- In scikit-learn, SVC unfortunately does not have a loss function parameter but LinearSVC does!

SVM and Multiple Classes

Doesn't extend so nicely. Still, we can do…

- One vs. One classification
  - If K > 2 classes, simply compute $\binom{K}{2}$ classifiers
  - Take test observation and tally up times assigned to each of K classes → Most frequent class assigned

- One vs. All classification
  - Fit K classifiers, each comparing one class to (K-1) classes
  - Take test observation and assign to each class we have highest confidence in ⇔ class for which $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}$ is highest