Classification: Generative Models (Naïve Bayes) and Support Vector Machines~~)~~

Siva Balakrishnan
Data Mining: 36-462/36-662

February 7th, 2018

6.6.3 of ESL (Naïve Bayes) Chapter 9 of ISL (SVMs) – mainly 9.1 today

Recap: Linear Discriminant Analysis

- For generative classifiers:
  - We estimate (prior) $\pi_k := \mathbb{P}(Y = k)$ and

$$f_k(x) = \mathbb{P}(X = x | Y = k).$$

  - We classify by:

$$\widehat{f}(x) = \arg\max_k f_k(x)\pi_k.$$

- For LDA we model:

$$f_k \sim N(\mu_k, \Sigma)$$

shared by all classes

- For QDA we model:

$$f_k \sim N(\mu_k, \Sigma_k)$$

Recap: Decision Boundaries in LDA

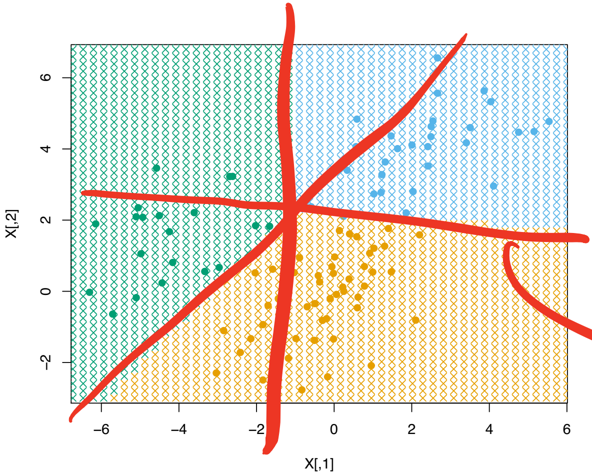
- By some algebra we saw that we could write the LDA decision rule as:

$$\widehat{f}(x) = \arg\max_k \Big[\underbrace{\log \pi_k - \frac{1}{2}\mu_k^T \Sigma^{-1} \mu_k + \mu_k^T \Sigma^{-1} x}_{\text{affine function of } x.} \Big]$$

$$\Downarrow$$

$$\delta_k(x)$$

- This leads to a picture like below:



$\delta_{blue}(x) = \delta_{orange}(x)$

Recap: Estimation in LDA

- We need to estimate the means for each class, and a single covariance matrix (common to all the classes).
- We do this via maximum likelihood (with minor adjustments) on the training data:

$$\widehat{\mu}_k = \frac{1}{n_k} \sum_{i=1, y_i=k}^{n} x_i$$

$$\widehat{\pi}_k = n_k / n$$

$$\widehat{\Sigma} = \frac{1}{n-K} \sum_{j=1}^{K} \sum_{i: y_i=j} (x_i - \hat{\mu}_j)(x_i - \hat{\mu}_j)^T$$

- Once we have estimated these we simply plug them in to find the decision rule we actually use:

$$\widehat{f}(x) = \underset{k}{\operatorname{argmax}} \left[\log \hat{\pi}_k - \frac{1}{2} \hat{\mu}_k \hat{\Sigma}^{-1} \hat{\mu}_k + \hat{\mu}_k \hat{\Sigma}^{-1} x \right].$$

Recap: The Mahalanobis Distance and LDA

$\arg\min_k d(x, \hat{\mu}_k)$

- Suppose that the classes were balanced, i.e. $\widehat{\pi}_k$ were the same for each k.
- Then our decision rule is equivalent to:

$$\widehat{f}(x) = \arg\max_k \left[-(x-\hat{\mu}_k)^T \hat{\Sigma}^{-1} (x-\hat{\mu}_k) \right]$$

- This quantity is known as the (squared) *Mahalanobis* distance between the point and the centroid. More generally:

$$d(x, y) = \sqrt{(x-y)^T \widehat{\Sigma}^{-1} (x-y)}$$

measures the distance standardized appropriately by the variances. Roughly, (and in 1D this is indeed true) it is measuring "how many standard deviations away from y is x?"

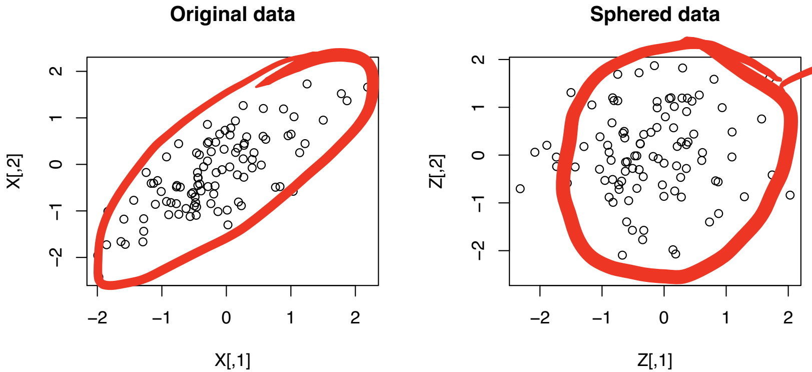
Recap: Sphering

- We could alternatively transform the data by creating:

$$\widetilde{x}_i = \widehat{\Sigma}^{-1/2} x_i.$$

- Then in the balanced case (when $\widehat{\pi}_k$ are equal) our rule is:

$$\widehat{f}(\widetilde{x}) = \underset{k}{\arg\min} \, \|\widetilde{x} - \widetilde{\mu}_k\|_2^2$$



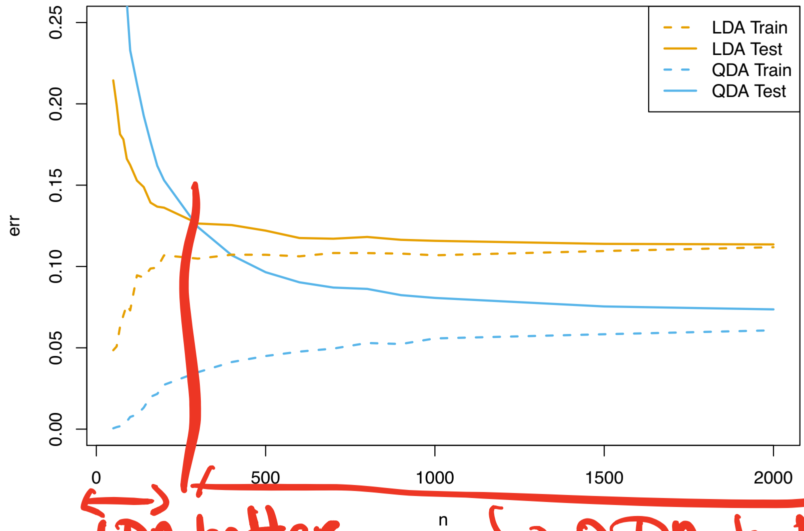
features are uncorrelated

Recap: Bias-Variance in LDA versus QDA

- Parameter counts: $Kp + \frac{p(p+1)}{2}$ $\quad X \in \mathbb{R}^p$.

QDA: $Kp + \frac{K p(p+1)}{2}$ → lower bias

- In typical cases: LDA has lower variance.



LDA better

QDA better

Variations on LDA: high dimensions

$O(p^2)$ may be too big.

Suppose the dimension gets even higher?

(1) $\Sigma = I$, nearest centroid classifier.

(2) Σ is diagonal $\rightarrow$ I only est variances.

What if I can't even estimate the group means well?

Regularize. $\rightarrow$ Nearest shrunken centroids.

$\Rightarrow$ $\text{Variance}(\hat{\Sigma}) \sim \dfrac{\#\text{ parameters in } \hat{\Sigma}}{n}$.

$$\Sigma = \begin{bmatrix} \ddots & & 0 \\ & \ddots & \\ 0 & & \ddots \end{bmatrix}$$

Large values here

Naive Bayes

Imagine that you have $n = 2000$ observations and $p = 1000$ features.

It will be incredibly hard to estimate $f_k = P(X = x|Y = k)$ well for any complicated model!

→ 1000 dimensional density for each class.

You need very strong "assumptions" on f_k (reducing parameters/variance)!

Naive Bayes *assumes* (well, models) that all of the components of $X = (X_1, \ldots, X_p)$ are *independent* (conditional on Y).

for each class

Naive Bayes

Naive Bayes models all of the components of $X = (X_1, \ldots, X_p)$ as *independent* (conditional on Y).

Under this independence assumption, the class distributions factor!

Naive Bayes

$$
\begin{aligned}
f_k(x) &= \underline{P(X = x|Y = k)} \\
&= \mathbb{P}(X_1 = x_1, X_2 = x_2, \ldots, X_p = x_p \,|\, Y = k) \\
&= \mathbb{P}(X_1 = x_1|Y = k) \times \mathbb{P}(X_2 = x_2|Y = k) \cdots \mathbb{P}(X_p = x_p|Y = k) \\
&= \prod_{j=1}^{p} \underbrace{\mathbb{P}(X_j = x_j|Y = k)}.
\end{aligned}
$$

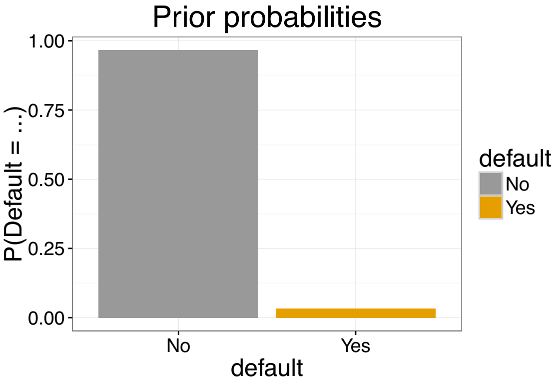
marginal distributions.

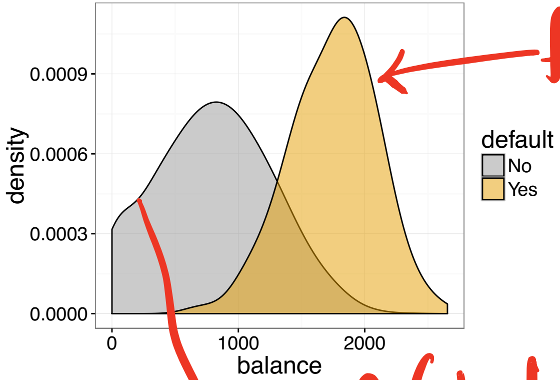
This means that we can just estimate the *univariate* distributions!

$$f_k^{(j)}(x_j) = P(X_j = x_j|Y = k)$$

Example: Default data from ISLR

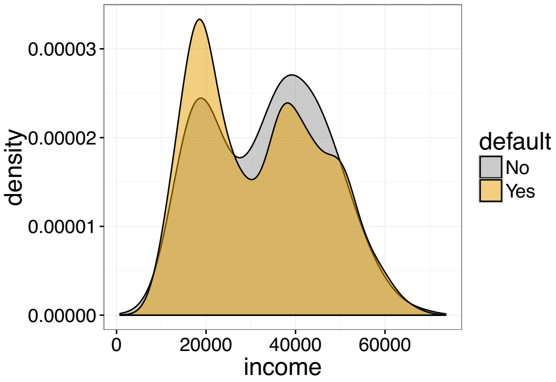
Default ~ Student/Not Balance Income

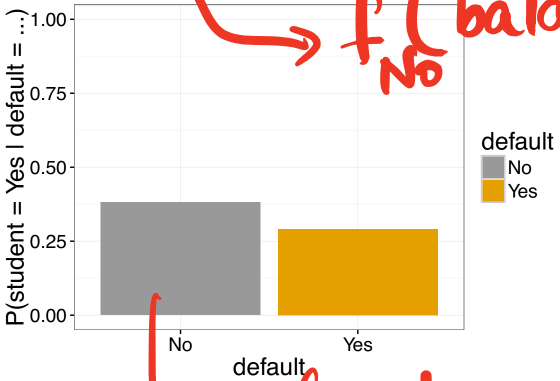




$f_{Yes}(\text{balance})$

$f_{No}(\text{balance})$





$\mathbb{P}(\text{student} = \text{yes} \mid \text{default} = \text{no})$.

We can easily calculate these simplified class distributions:

$\hat{f}(X \mid y = \text{Yes})$

$$\hat{f}_{\text{Yes}} = \hat{f}_{\text{Yes}}(\text{income})\hat{f}_{\text{Yes}}(\text{balance})\hat{f}_{\text{Yes}}(\text{student})$$

$$\hat{f}_{\text{No}} = \hat{f}_{\text{No}}(\text{income})\hat{f}_{\text{No}}(\text{balance})\hat{f}_{\text{No}}(\text{student})$$

$\hat{f}(X \mid y = \text{No})$.

And then plug them into the Bayes classifier formula, just like we did for LDA

$$\mathbb{P}(\text{default}|i, b, s) = \frac{\hat{\pi}_{\text{Yes}}\hat{f}_{\text{Yes}}(i, b, s)}{\hat{\pi}_{\text{Yes}}\hat{f}_{\text{Yes}}(i, b, s) + \hat{\pi}_{\text{No}}\hat{f}_{\text{No}}(i, b, s)}$$

not linear in x.

Say default is 0/No

$\hat{f}_{No}(x)\,\hat{\pi}_{no} \geq f_{yes}(x)\,\hat{\pi}_{yes}$.

Gaussian Naive Bayes

- When the covariates are all continuous, one version of the Naïve Bayes classifier assumes that:

univariate Gaussian

$$f_k(X_i) \sim N(\mu_{ik}, \sigma_i^2),$$

  i.e. that each feature is (univariate) Gaussian with common variance across classes. This is called the *Gaussian Naïve Bayes* classifier.
- We know this as a special case of another classifier? } LDA with diagonal covariance matrix.
- What can we say about the decision boundary of Gaussian Naïve Bayes (say in the two-class setting)?

decision bdy is linear.

Naive Bayes

Naive Bayes scales well to problems with very large p. We only need enough data to estimate each of the marginal distributions well.

It also allows you to have a flexible choice of models for each of the univariate distributions.

However, Naive Bayes cannot capture *interactions* between the features within each class!

LDA and QDA are able to incorporate these feature interactions, at the cost of needing to estimate them.

Support Vector Machines: Hyperplanes Review

For the rest of today we will consider binary classification problems where $y_i \in \{-1, 1\}$ (instead of $y_i \in \{0, 1\}$). Several classifiers look at $x_i^T\beta + \beta_0$, and assign

LDA, logistic, GNB

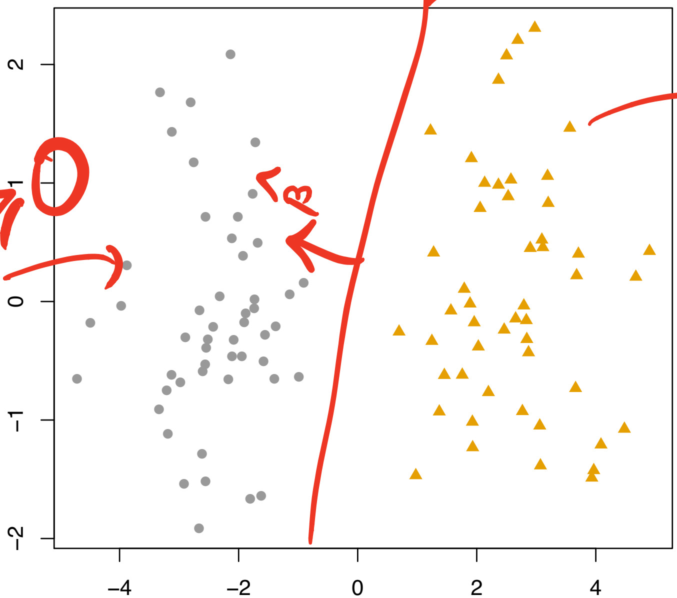
$$\widehat{f}(x) = \begin{cases} 1 & x^T\widehat{\beta} + \widehat{\beta}_0 > 0 \\ -1 & x^T\widehat{\beta} + \widehat{\beta}_0 < 0 \end{cases} = \text{sign}(x^T\widehat{\beta} + \widehat{\beta}_0)$$

$x^T\hat{\beta} + \beta_0 = 0.$

How should we think about this geometrically?

$x^T\hat{\beta} + \beta_0 < 0$

$x^T\hat{\beta} + \beta_0 > 0$



Linearly Separable Data

- If our data is linearly separable then there is some (β, β_0) such that:

$$\text{if } \beta^T x_i + \beta_0 > 0 \quad y_i = +1$$

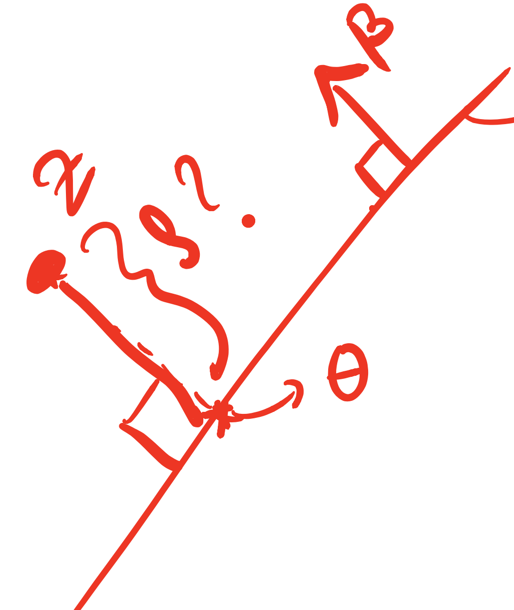
$$\& \text{ if } \beta^T x_i + \beta_0 < 0 \quad y_i = -1.$$

- Equivalently:

$$y_i(\beta^T x_i + \beta_0) > 0 \quad \forall i.$$

More Hyperplanes Review

- How far is a point from a hyperplane?



$\beta^T x + \beta_0 = 0.$

$10\beta^T x + 10\beta_0 = 0$

Standardize by saying:

$$\sum_{j=1}^{p} \beta_j^2 = 1.$$

Punchline: $\delta = |\beta^T z + \beta_0|.$

$$\theta = z - (z^T\beta + \beta_0)\beta$$

$$\beta^T\theta + \beta_0 = \beta^T z - (z^T\beta + \beta_0)\underbrace{\beta^T\beta}_{1} + \beta_0.$$

$$= 0.$$

$$\delta = \|\theta - z\| = \|(z^T\beta + \beta_0)\beta\|$$

$$= |z^T\beta + \beta_0| \, \underbrace{\|\beta\|_2}_{1}$$

$$= |z^T\beta + \beta_0|.$$

Building a linear classifer

Suppose I want to build a nice, linear classifier $\mathrm{sign}(x^T\beta + \beta_0)$. How should I choose (β, β_0)?

1. I could build a model of each cloud of points, and classify to the best model

2. I could model probability $P(Y = 1|X)$ with a linear model,

$$P(Y = 1|X = x) = \frac{1}{1 + e^{-x^T\beta}}$$

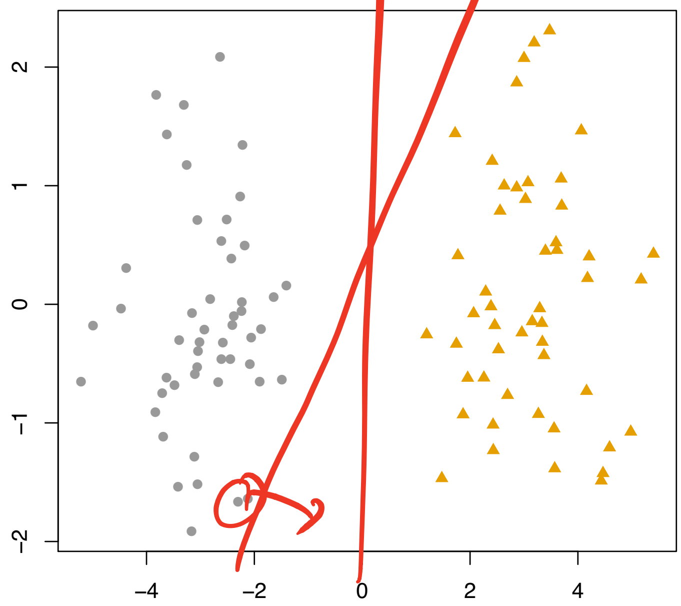
Logistic regression

3. I could just try to draw a line down the middle.

Maximum margin classifier: a special case

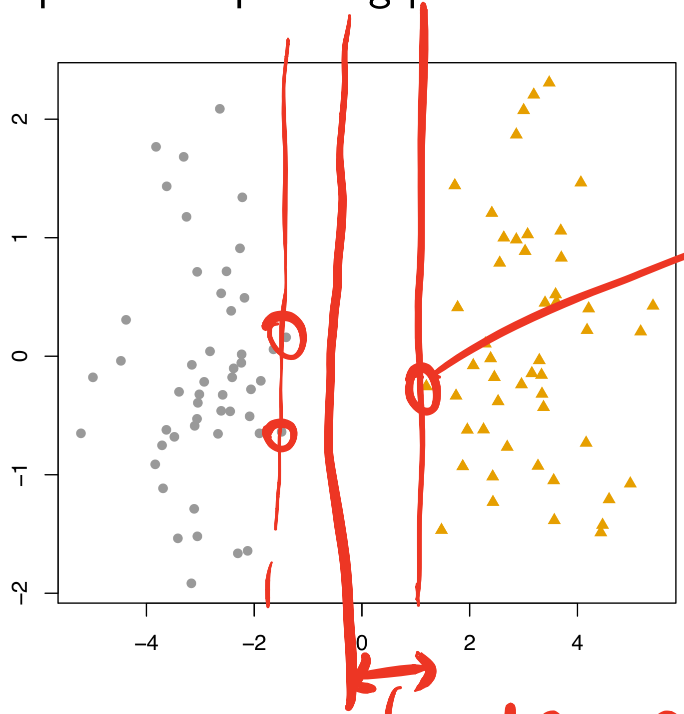
Where should we put our separating plane?

maybe bad



Maximum margin classifier: a special case

Where should we put our separating plane?



Support vectors.

margin.

can throw away all non support vectors.

Maximum margin classifier: a special case

M, β, β_0 are all variables.

Maximize M → margin

Subject to $y_i(x_i^T\beta + \beta_0) \geq M$ and $\sum_{j=1}^{p} \beta_j^2 = 1$

with this standardize

↳ $|x_i^T \beta + \beta_0|$ → distance between x_i & hyperplane

This is not based on model assumptions! This is just a nice idea of what "draw a separating line" should look like.

SVM in sep. case

Note that the plane only depends on the points right at the boundary. The other points could move around and nothing would change.

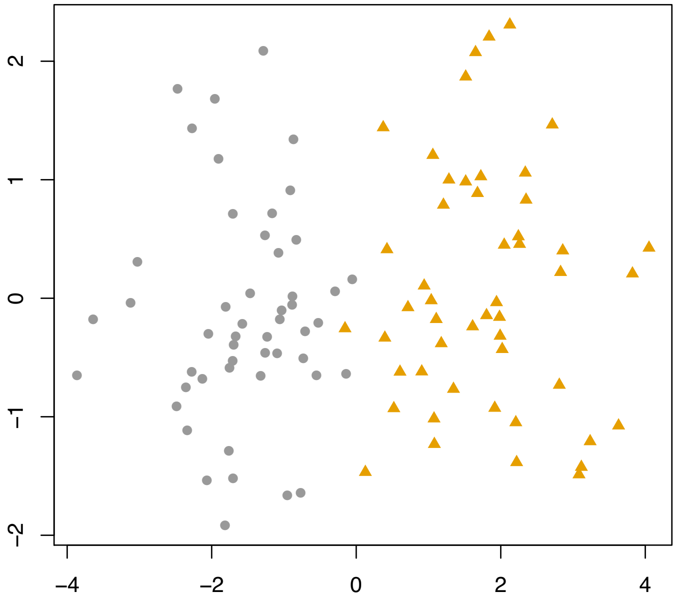
However, this is only defined if we have nicely separable groups! That seems a bit wishful.

→ if not sep, then cannot solve

allow "slack".

Maximum margin classifier: a special case

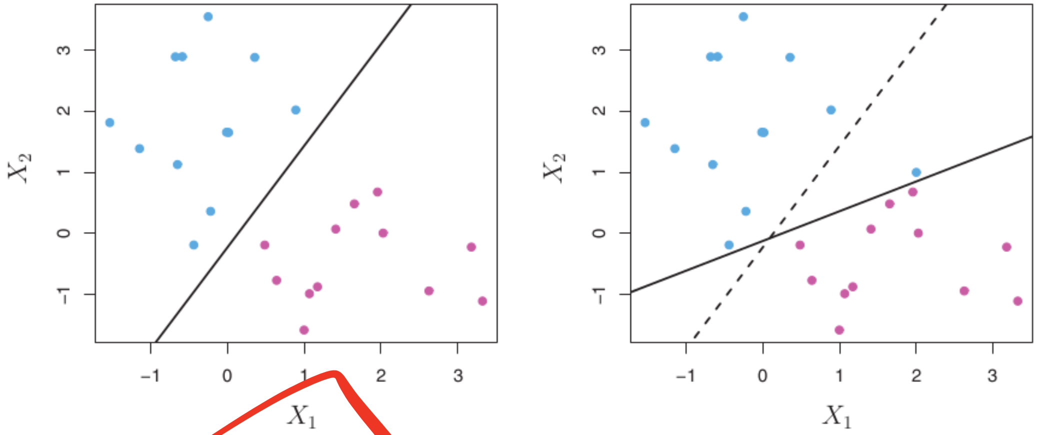
Now what??



Support vector classifier

We need to relax the notion of a margin, in case the groups cannot actually be separated. We introduce a notion of a soft margin which allows some violations.

This has unexpected benefits! Strict margins give the boundary points too much influence. Now we can tune the variance of the boundary.



has high variance.

(ISL pg.345)

Support vector classifier

$\varepsilon_i \to$ slack vars.

Maximize M

subject to $\sum_{j=1}^{p} \beta_j^2 = 1$, $\varepsilon_i \geq 0$, $\sum_{i=1}^{n} \varepsilon_i \leq C$

$C \to$ tuning param.

$y_i(x_i^T \beta + \beta_0) \geq M(1 - \varepsilon_i)$

$\varepsilon_i > 0$, margin for $i < M$

$\varepsilon_i > 1$. then misclassifi

- Parameter C determines "softness" of the margin. Big C makes it easier to cross. In particular, no more than C observations cross because...

$\to$ fewer than C training error.

- Variable ε_i encodes point location: $\varepsilon_i = 0$ outside margin, $\varepsilon_i > 0$ inside margin, $\varepsilon_i > 1$ across boundary.

If $\varepsilon_i > 1$, $y_i(x_i^T \beta + \beta_0)$ is negative