Classification for the Car Crashes Analysis

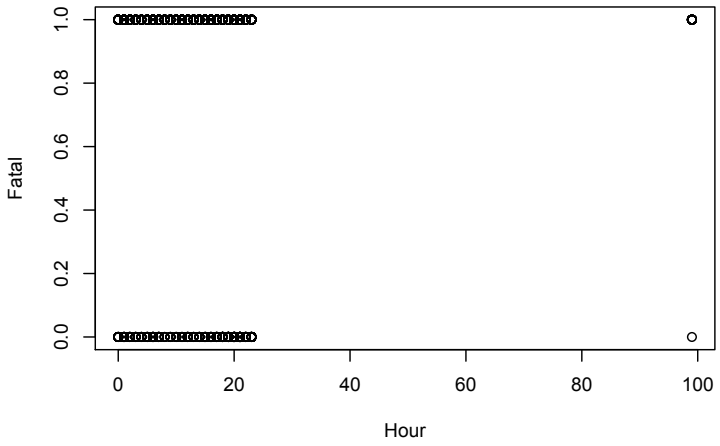
Car Crashes Data

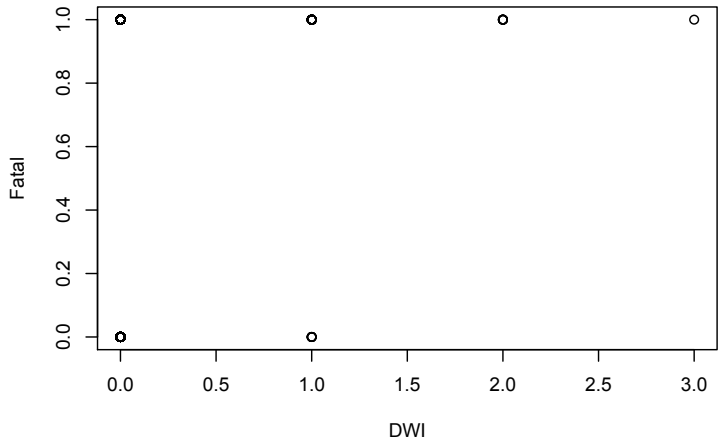
What is our goal with the Car Crashes data?

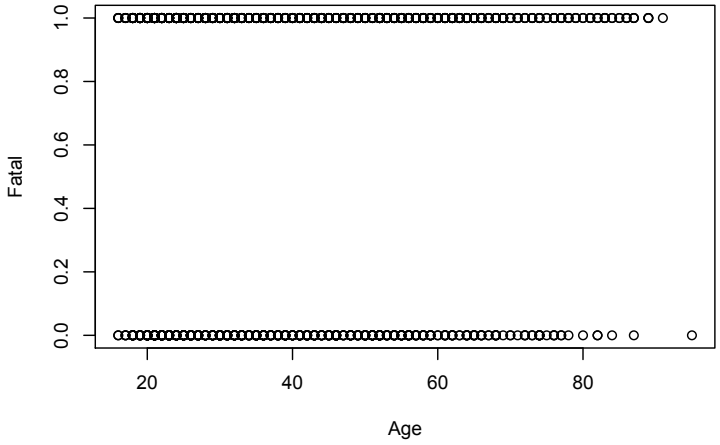
1. Infer the relationship between variables and the probability of a fatality in the vehicle.

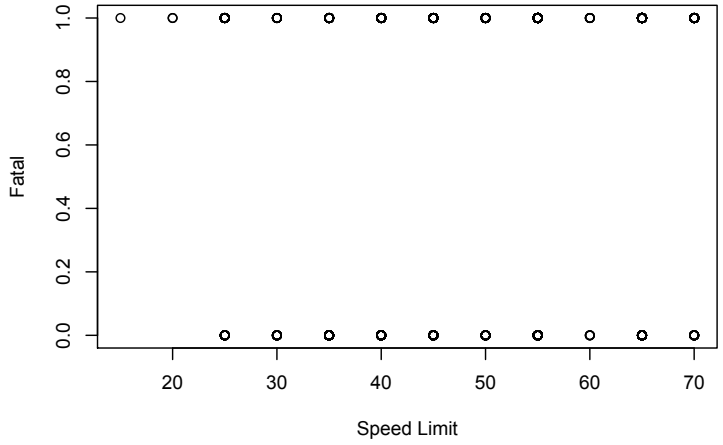
Why is this goal important?

- Enact policy to save lives.


Fatal
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
Hour


Fatal
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
DWI


Fatal
0.0
0.2
0.4
0.6
0.8
1.0
20
40
60
80
Age


Fatal
0.0
0.2
0.4
0.6
0.8
1.0
20
30
40
50
60
70
Speed Limit

	No Fatalities	Fatalities
DOT-Helmet	1	21
Lap Belt Only Used	5	6
No Helmet	0	14
None Used-Motor Vehicle Occupant	12	209
Not Applicable	2	4
Not Reported	4	2
Other Helmet	4	148
Shoulder and Lap Belt Used	456	429
Shoulder Belt Only Used	2	5
Unknown	18	46

Car Crashes Data

What are the challenges with the Car Crashes data?

1. 0/1 Response.
2. Lots of variables.

This unit is all about dealing with categorical responses.
This is referred to as classification.

Possible Approaches:

1. Logistic Regression
2. Probit Regression
3. Discriminant Analysis
4. K-Nearest Neighbors
5. Support Vector Machines

Why Not Linear Regression?

Why can't we just define:

$$Y_i = \begin{cases} 1 & \text{if Fatalities} \\ 0 & \text{otherwise} \end{cases}$$

then use:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}?$$

Because:

1. Predictions $\mathbf{x}_0'\hat{\boldsymbol{\beta}} \notin \{0, 1\}$.
2. Its certainly not-linear.

Logistic Regression

Thinking about linear regression, we have:

$$\mathbf{Y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I})$$

so, when data are continuous we use a Gaussian distribution and are interested in regressing the mean.

1. What's an appropriate distribution when $Y_i \in \{0, 1\}$? Bernoulli Distribution: $f(y_i) = p^{y_i}(1-p)^{1-y_i}$
2. When we use a Bernoulli distribution, what are we interested in?

$$p = \text{Prob}(Y_i = 1)$$

Logistic Regression

So, a regression model for categorical data would be:

$$p = \text{Prob}(Y = 1) = \mathbf{x}_i'\boldsymbol{\beta}$$

Problem: p has to be between 0 and 1. How do we constrain our regression to be between 0 and 1?

Logistic Regression Model: (Generalized Linear Model)

Odds Ratio

$$Y_i \overset{ind}{\sim} \text{Bern}(p_i)$$

$$\log\left(\frac{p_i}{1-p_i}\right) = \mathbf{x}_i'\boldsymbol{\beta}$$

Logistic Function

$$\Rightarrow p_i = \frac{\exp\{\mathbf{x}_i'\boldsymbol{\beta}\}}{1+\exp\{\mathbf{x}_i'\boldsymbol{\beta}\}}$$

Logit Transform

Logistic Regression

Logistic Regression Model:

$$\log\left(\frac{p_i}{1-p_i}\right) = \mathbf{x}_i'\boldsymbol{\beta}$$

How do we interpret β_j?

1. For every unit increase in x_j, the log-odds ratio increases by β_j.
2. Just interpret the sign: If $\beta_j > 0$, then p_i increases as x_j increases.
3. $100 \times (\exp\{\beta_j\} - 1)$: percent increase in the odds ratio.
4. Draw a picture.

Logistic Regression

How do we estimate $\boldsymbol{\beta}$?

- Maximum the likelihood:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} f(y_i \mid \boldsymbol{\beta}) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{y_i}$$

$$= \prod_{i=1}^{n} \left(\frac{\exp\{\mathbf{x}_i'\boldsymbol{\beta}\}}{1+\exp\{\mathbf{x}_i'\boldsymbol{\beta}\}} \right)^{y_i} \left(1 - \frac{\exp\{\mathbf{x}_i'\boldsymbol{\beta}\}}{1+\exp\{\mathbf{x}_i'\boldsymbol{\beta}\}} \right)^{1-y_i}$$

$$= \prod_{i=1}^{n} \left(\frac{\exp\{\mathbf{x}_i'\boldsymbol{\beta}\}}{1+\exp\{\mathbf{x}_i'\boldsymbol{\beta}\}} \right)^{y_i} \left(\frac{1}{1+\exp\{\mathbf{x}_i'\boldsymbol{\beta}\}} \right)^{1-y_i}$$

- In R Code: glm(y~X,family=binomial)

Logistic Regression

How do we get standard errors (uncertainties) associated with the MLE $\hat{\boldsymbol{\beta}}$?

- Asymptotics:

$$\hat{\boldsymbol{\beta}} \xrightarrow{d} \mathcal{N}(\boldsymbol{\beta}, \mathbf{I}^{-1}(\boldsymbol{\beta}))$$

glm will calculate these for you. Confidence intervals can be computed by:

```
the.glm <- summary(glm(y~X,family=binomial))
upper <- the.glm$coef[,1] + qnorm(0.975)*the.glm$coef[,2]
lower <- the.glm$coef[,1] - qnorm(0.975)*the.glm$coef[,2]
```

- Bootstrap it!
- Be a Bayesian!

Logistic Regression

How do we get predictions?

- We predict the probabilities:

$$p_i = \frac{\exp\{\mathbf{x}_0'\hat{\boldsymbol{\beta}}\}}{1 + \exp\{\mathbf{x}_0'\boldsymbol{\beta}\}}$$

How do we classify from these probabilities?

1. Classify based on highest probability (Bayes Classifier).
2. Probably want to consider cost/benefits to set a cut-off probability.

Logistic Regression

How do we assess accuracy of our prediction?

- Cross-Validation

Confusion Matrix

	Predicted Yes's	Predicted No's
True Yes's	10	2
True No's	4	12

Error Rate

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i) = \text{Percent Misclassified}$$

Logistic Regression

How do we assess accuracy of our prediction?

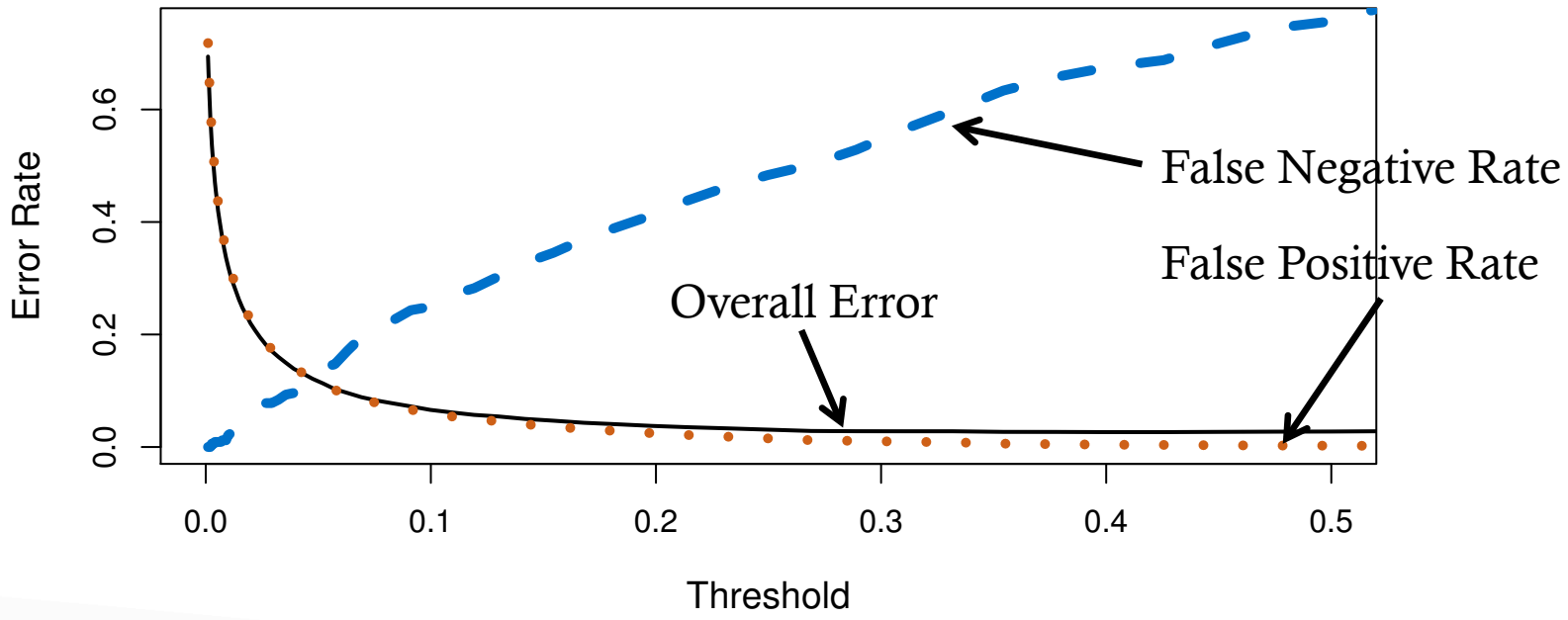
	Predicted Yes's	Predicted No's
True Yes's	10	2
True No's	4	12

- Sensitivity: Percent of True Positives (10/12)
- Specificity: Percent of True Negatives (12/16)
- Positive Predictive Value: % Correctly Predicted Yes's (10/14)
- Negative Predictive Value: % Correctly Predicted No's (12/14)

Logistic Regression

How do we assess accuracy of our prediction?

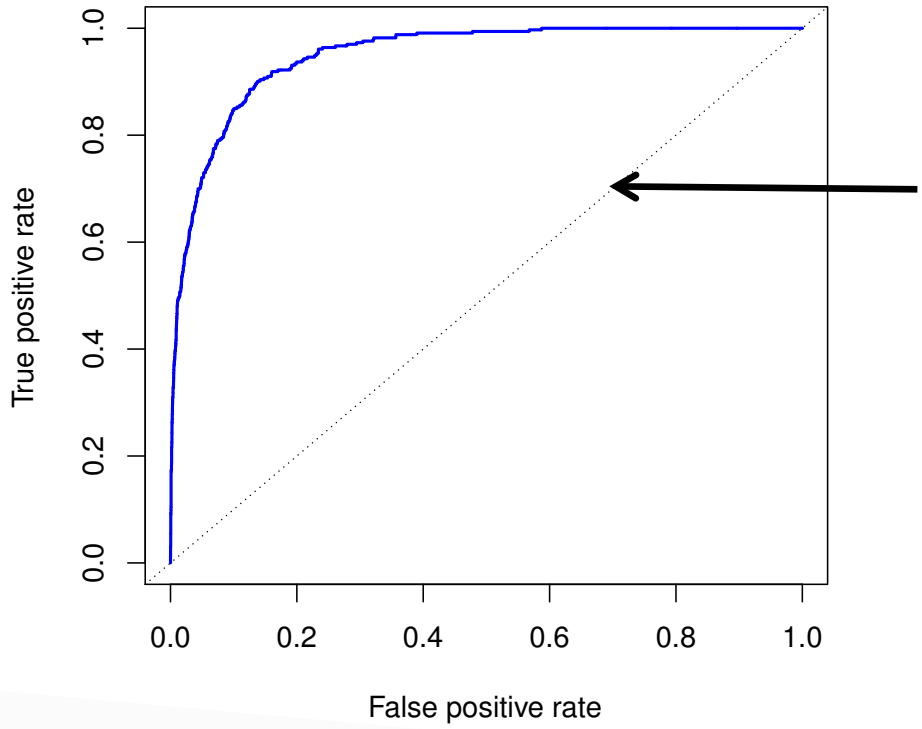
Classification Threshold Revisited: What probability should we use to classify observations? It Depends!



Logistic Regression

How do we assess accuracy of our prediction?

ROC (Receiver Operating Curves): Compares sensitivity to false positive rates for many thresholds.



"Coin Flip" Rate

Summarize an ROC curve by the area under the curve (AUC).

Logistic Regression

How do we do variable selection in logistic regression?

The same way we do in linear regression!

$$\text{AIC} = -2\log(\text{Like}) + 2P$$

$$\text{BIC} = -2\log(\text{Like}) + P\log(n)$$

$$\text{Training Set Misclassification}$$

$$\text{Cross Validation Misclassification}$$

Logistic Regression

Do we have to worry about collinearity in logistic regression?

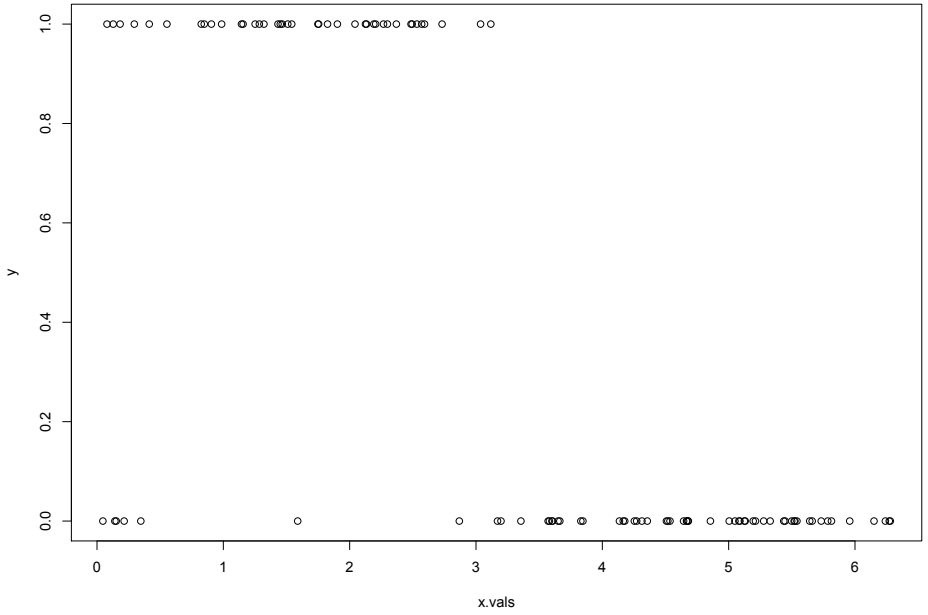
Yes!

What can we do if we have multicollinearity?

1. Orthogonalize the variables.
2. Principal components analysis (this is portable to GLMs).
3. LASSO or Ridge - glmnet(...,family=binomial).
4. Strong prior correlations.

Logistic Regression

What do we do if there is non-linearity (non-monotonicity)?



1. Splines
2. Smoothing splines - gam(...,family=binomial)

Probit Regression

Probit Regression Model:

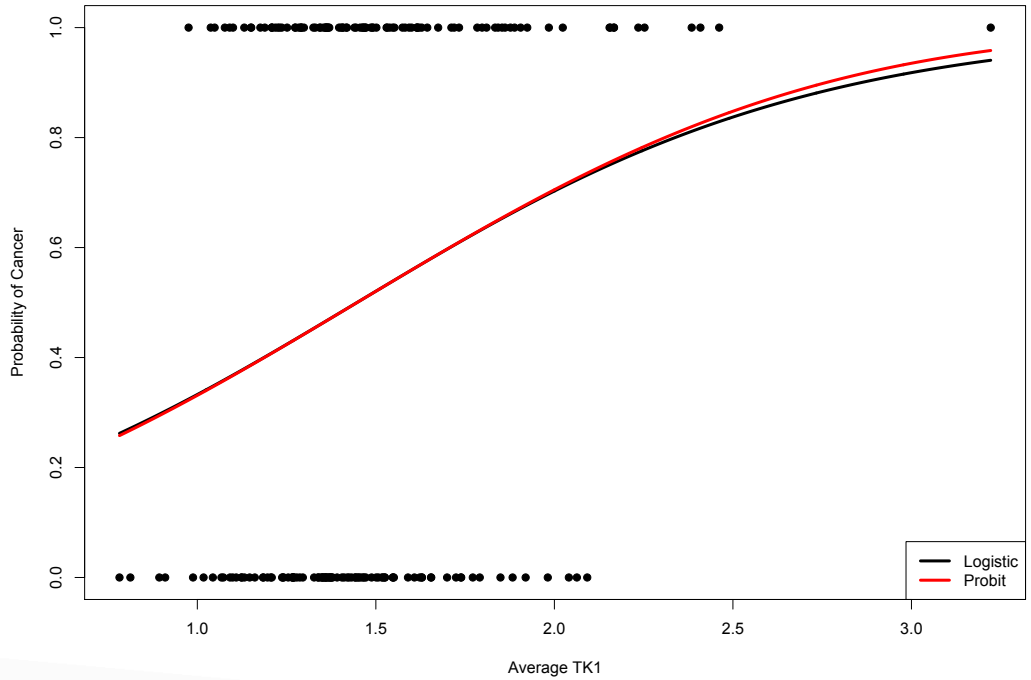
$$Y_i \overset{ind}{\sim} \text{Bern}(p_i)$$
$$p_i = \Phi(\mathbf{x}_i'\boldsymbol{\beta})$$
$$\Phi = \text{Normal CDF}$$

In R: glm(y~X,family=binomial(link="probit"))

Probit Regression

Logistic vs. Probit Regression : Is there a difference?

Coefficients will be different, but probability curve will be about the same.



	$\hat{\beta}_1$
Logistic	1.556
Probit	0.976

Probit Regression

Logistic vs. Probit Regression :

	Advantages	Disadvantages
Logistic	Interpretability	No conjugacy
Probit	Conjugacy in Bayesian Model	Can only interpret sign of coefficients.

Discriminant Analysis

Bayes Theorem for Classification:

New Idea: Assume $\mathbf{X}$ is random and factor the joint distribution $[Y, \mathbf{X}] = [\mathbf{X} \mid Y][Y]$.

$$\text{Prior Probabilities: } \text{Prob}(Y = 1) = \pi_1$$

$$\text{Conditional Dist: } [\mathbf{X} \mid Y = 1] \sim \mathcal{N}_p(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$

$$[\mathbf{X} \mid Y = 0] \sim \mathcal{N}_p(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

Using Bayes Theorem:

$$[Y = 1 \mid \mathbf{X}] = \frac{\texttt{dmvnorm}(\mathbf{X} \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\pi_1}{\texttt{dmvnorm}(\mathbf{X} \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\pi_1 + \texttt{dmvnorm}(\mathbf{X} \mid \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)(1 - \pi_1)}$$

$$\texttt{dmvnorm}(\mathbf{X} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^P |\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{X} - \boldsymbol{\mu})\right\}$$

Discriminant Analysis

Discriminant Analysis Classification: Assign Y to the class that maximizes,

$$\max_{i \in \{0,1\}} \texttt{dmvnorm}(\mathbf{X} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\pi_i$$

$$\max_{i \in \{0,1\}} -\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\mathbf{X} - \boldsymbol{\mu}_i)' + \log(\pi_i)$$

Note: Technically, this is "quadratic" discriminant analysis. "Linear" discriminant analysis assumes $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$.

Discriminant Analysis

Discriminant Analysis Classification:

Issues:

1. What do we use for $\pi_0, \pi_1, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_0, \text{and } \boldsymbol{\Sigma}_1$?

$$\hat{\pi}_0 = \frac{1}{N}\sum_{i=1}^{N} I(y_i = 0) \qquad \hat{\boldsymbol{\mu}}_1 = \frac{1}{N_1}\sum_{i:y_i=1} \mathbf{X}_i$$

$$\hat{\pi}_1 = \frac{1}{N}\sum_{i=1}^{N} I(y_i = 1) \qquad \hat{\boldsymbol{\Sigma}}_0 = \frac{1}{N_0 - 1}\sum_{i:y_i=0} (\mathbf{X}_i - \hat{\boldsymbol{\mu}}_0)(\mathbf{X}_i - \hat{\boldsymbol{\mu}}_0)'$$

$$\hat{\boldsymbol{\mu}}_0 = \frac{1}{N_0}\sum_{i:y_i=0} \mathbf{X}_i \qquad \hat{\boldsymbol{\Sigma}}_1 = \frac{1}{N_1 - 1}\sum_{i:y_i=0} (\mathbf{X}_i - \hat{\boldsymbol{\mu}}_1)(\mathbf{X}_i - \hat{\boldsymbol{\mu}}_1)'$$

$$N_0 = \text{Number of } Y_i = 0, N_1 = \text{Number of } Y_i = 1$$

Discriminant Analysis

Discriminant Analysis Classification:

Issues:

2. Makes NO SENSE when you have categorical predictors! You would never use a normal distribution for a categorical variable. Thought question: Can you think of ways to adjust discriminant analysis to better utilize categorical variables?
3. There is no measure of uncertainty in your prediction.

K-Nearest Neighbors

K-Nearest Neighbors Algorithm (Non-parametric):

To classify a point which has covariates $\mathbf{x}_0$,

1. Find the K-nearest neighbors according to some distance $d_{0i} = d(\mathbf{x}_0, \mathbf{x}_i)$.
2. Approximate,

$$\text{Prob}(Y = 1 \mid \mathbf{x}_0) \approx \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = 1)$$

where $\mathcal{N}_0$ is the set of K nearest points.

3. Classify according to majority vote.

K-Nearest Neighbors

Issues:

1. What distance, $d_{0i} = d(\mathbf{x}_0, \mathbf{x}_i)$, should we use?
   - Euclidean – does this make sense for categorical variables?
   - Mahalanobis - $\sqrt{(\mathbf{x}_0 - \mathbf{x}_i)'\mathbf{S}^{-1}(\mathbf{x}_0 - \mathbf{x}_i)}$
2. What value should we use for K?
   - Use Cross-validation

Support Vector Machines

Building Block #1: Separating Hyperplanes

Hyperplane: A flat subset with dimension $P - 1$.

- In $\mathbb{R}^1$: a point.
- In $\mathbb{R}^2$: a line.
- In $\mathbb{R}^3$: a plane

Mathematically: $\mathcal{H}_P = \{\mathbf{x} : \beta_0 + \mathbf{x}'\boldsymbol{\beta} = 0\}$.

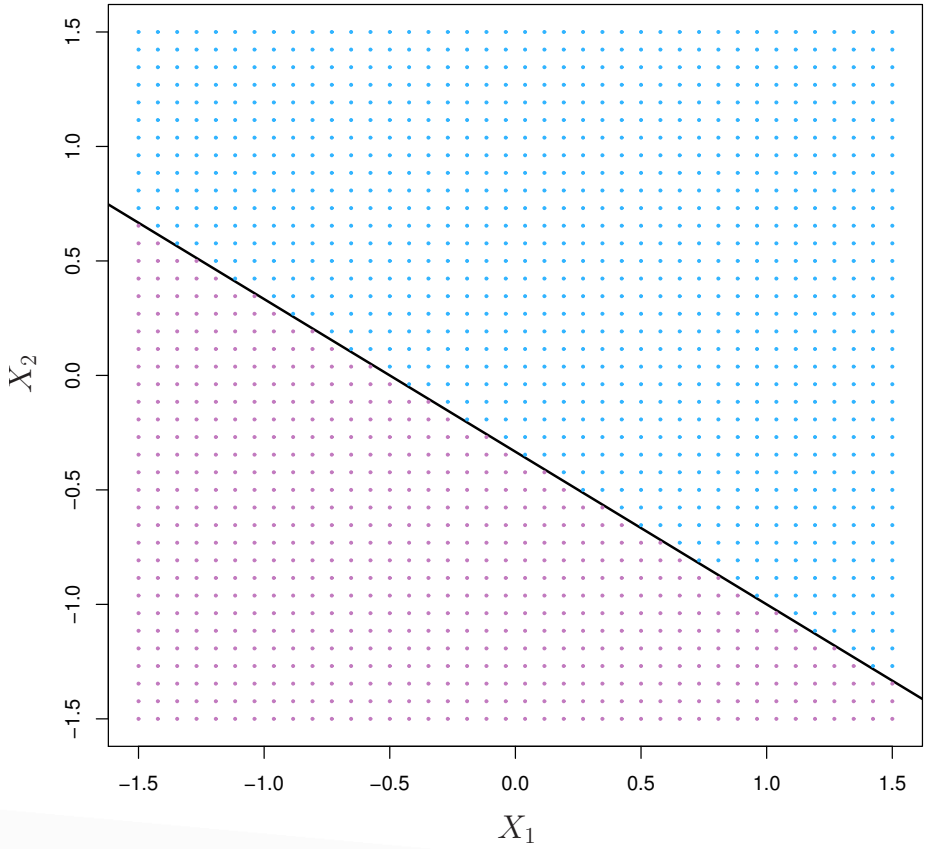
Divides the P -dimensional space in half. For example, if

$$\beta_0 + \mathbf{x}_0'\boldsymbol{\beta} > 0$$

then the point $\mathbf{x}_0$ is "above" the hyperplane. Otherwise, its "below" the hyperplane.

Support Vector Machines

Building Block #1: Separating Hyperplanes



For $P = 2$, let $\mathbf{x} = (x, y)$.

$$0 = \beta_0 + \beta_1 x + \beta_2 y$$

$$\Rightarrow y = -\frac{\beta_0}{\beta_2} - \frac{\beta_1}{\beta_2} x$$

Also, if

$$0 < \beta_0 + \beta_1 x + \beta_2 y$$

$$\Rightarrow y > -\frac{\beta_0}{\beta_2} - \frac{\beta_1}{\beta_2} x$$

so you're "above" the line.

Support Vector Machines

Building Block #1: Separating Hyperplanes

Big Idea: Find a hyperplane that separates the different classes.

Let $y_i \in \{-1, 1\}$, rather than $y_i \in \{0, 1\}$. We want to find a hyperplane such that,

$$y_i(\beta_0 + \mathbf{x}_i'\boldsymbol{\beta}) > 0 \ \forall i.$$

The classification rule becomes:

$$\hat{y}_0 = \text{sign}(\beta_0 + \mathbf{x}_0'\boldsymbol{\beta})$$

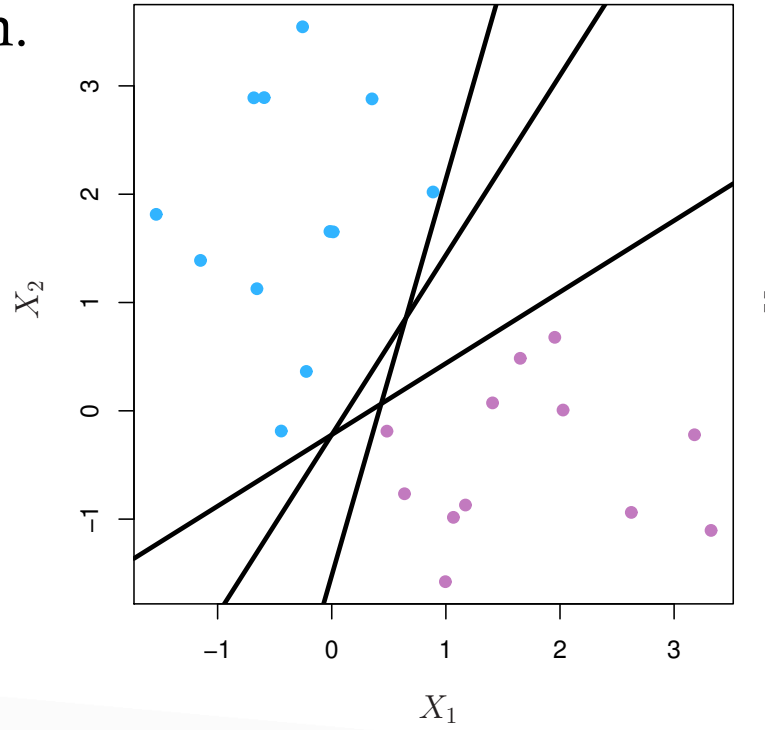
and, magnitude of $\beta_0 + \mathbf{x}_0'\boldsymbol{\beta}$ becomes how far away a point is from the hyperplane. The function $\beta_0 + \mathbf{x}_0'\boldsymbol{\beta}$ is referred to as the classifier (or classifying function).

Support Vector Machines

Building Block #1: Separating Hyperplanes

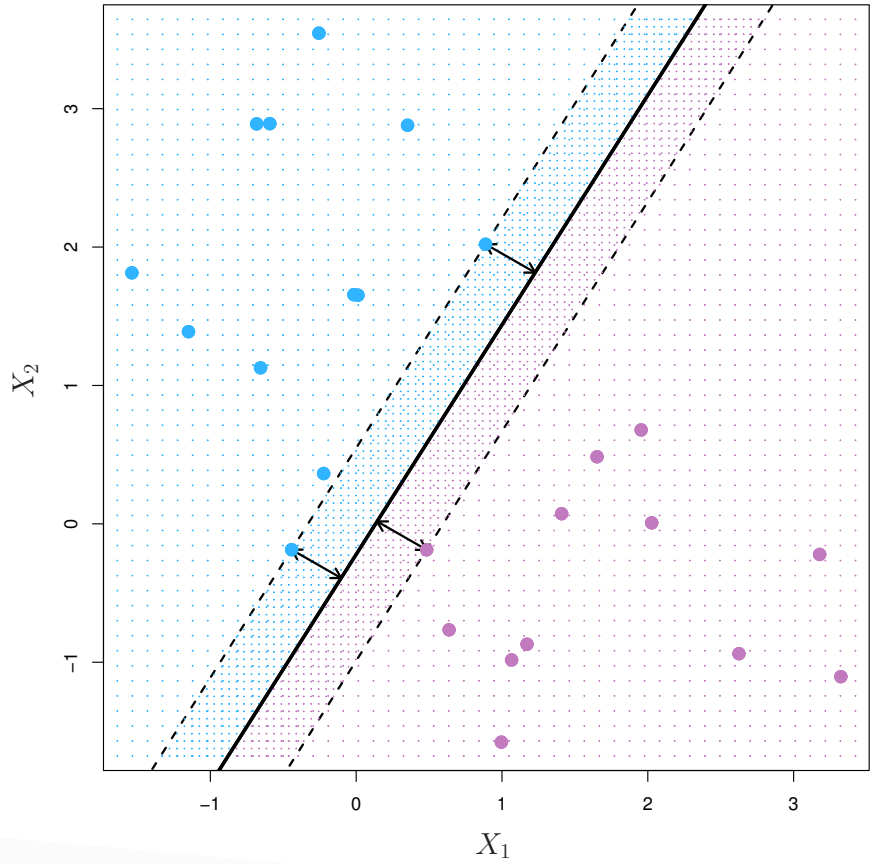
Issues with finding separating hyperplanes:

1. If such a plane does exists, there are infinitely many of them.



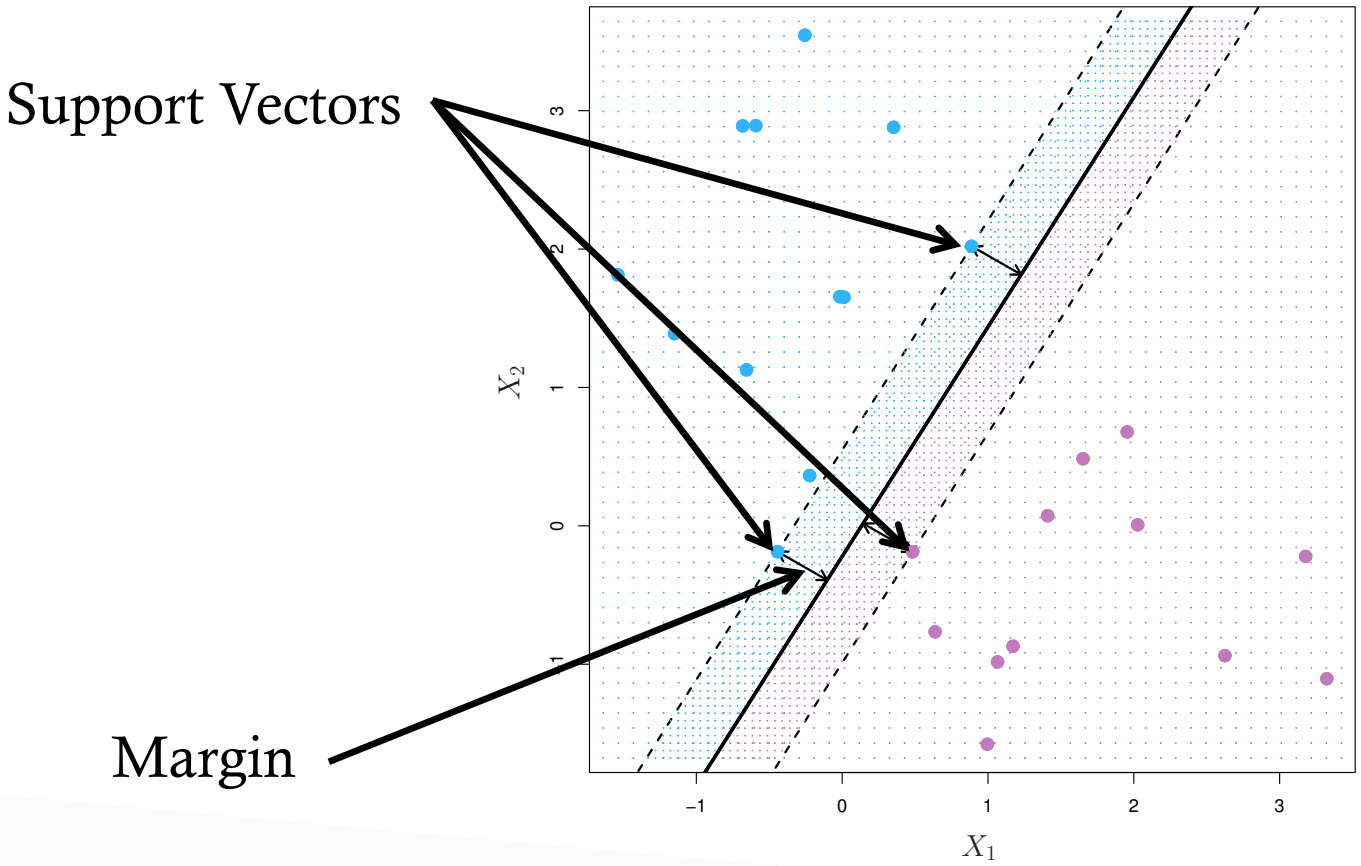
Support Vector Machines

Maximal Margin Hyperplane: Plane that maximizes orthogonal distance from observations to plane.



Support Vector Machines

Maximal Margin Hyperplane: Plane that maximizes orthogonal distance from observations to plane.

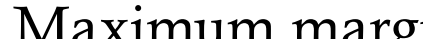


Support Vector Machines

Maximal Margin Hyperplane: Plane that maximizes orthogonal distance from observations to plane.

Mathematically, find the hyperplane that

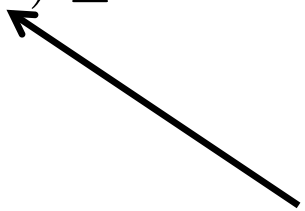
$$\max_{\beta_0,\ldots,\beta_P} M$$



Maximum margin

$$\text{subject to} \sum_{p=1}^{P} \beta_p^2 = 1$$

$$y_i(\beta_0 + \mathbf{x}_i'\boldsymbol{\beta}) \geq M \ \forall \ i.$$



Ensures points are on the "right" side of the hyperplane.

Support Vector Machines

Building Block #1: Separating Hyperplanes

Issues with finding separating hyperplanes:

1. If such a plane does exists, there are infinitely many of them – use maximum margin hyperplane.
2. Separating hyperplanes, essentially, don't exist for real data.

Support Vector Machines

Building Block #2: Support Vector Classifier

Big Idea: Find a hyperplane that "almost" separates the classes.

$$\max_{\beta_0,\ldots,\beta_P} M$$

$$\text{subject to} \sum_{p=1}^{P} \beta_p^2 = 1$$

$$y_i(\beta_0 + \mathbf{x}_i'\boldsymbol{\beta}) \geq M(1 - \epsilon_i)$$

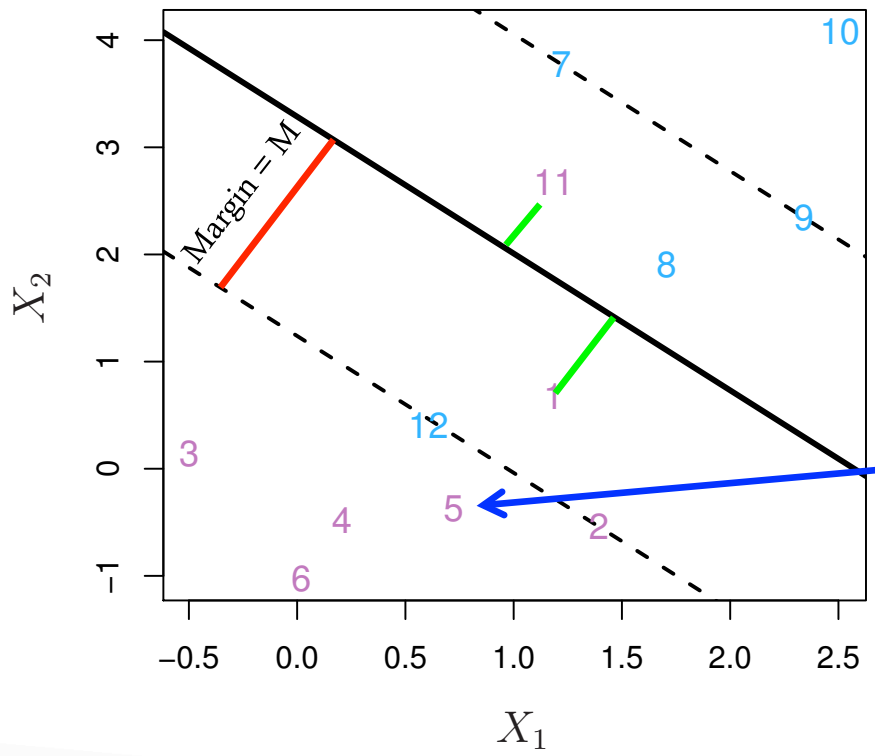
$$\epsilon_i \geq 0, \sum_{i=1}^{N} \epsilon_i \leq C$$

"Slack Variables"

Tuning Parameter

Support Vector Machines

Building Block #2: Support Vector Classifier



Purple: $y_i = -1$

Blue: $y_i = +1$

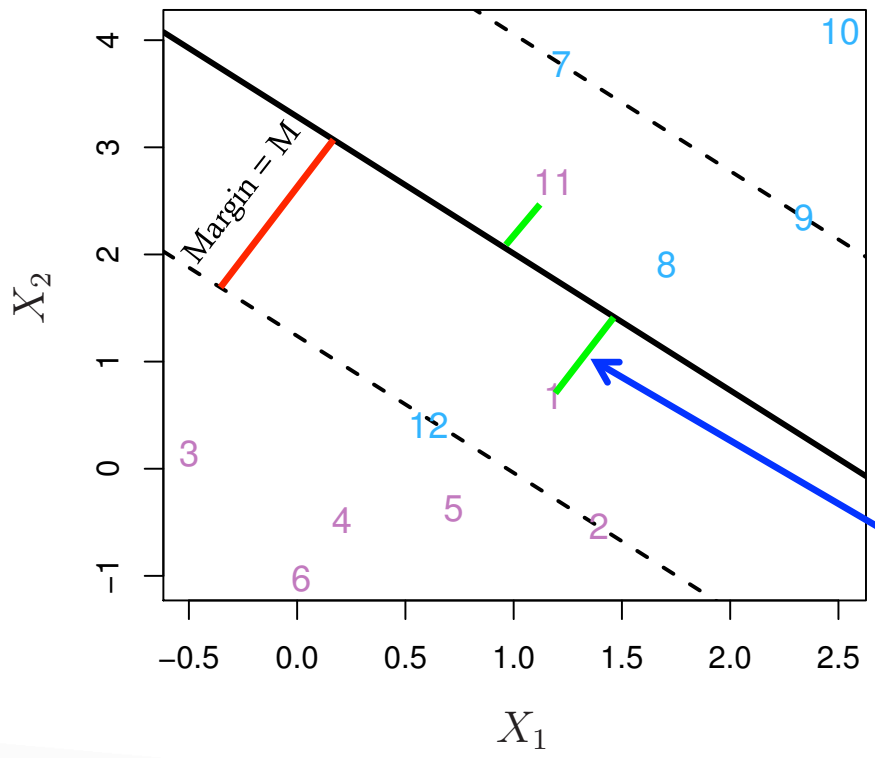
$$y_i(\beta_0 + \mathbf{x}_i'\boldsymbol{\beta}) \geq M(1 - \epsilon_i)$$

Slack Variables - ϵ_i

$\epsilon_i = 0 \Rightarrow$ Outside of Margin

Support Vector Machines

Building Block #2: Support Vector Classifier



$$\text{Purple: } y_i = -1$$

$$\text{Blue: } y_i = +1$$

$$y_i(\beta_0 + \mathbf{x}_i'\boldsymbol{\beta}) \geq M(1 - \epsilon_i)$$

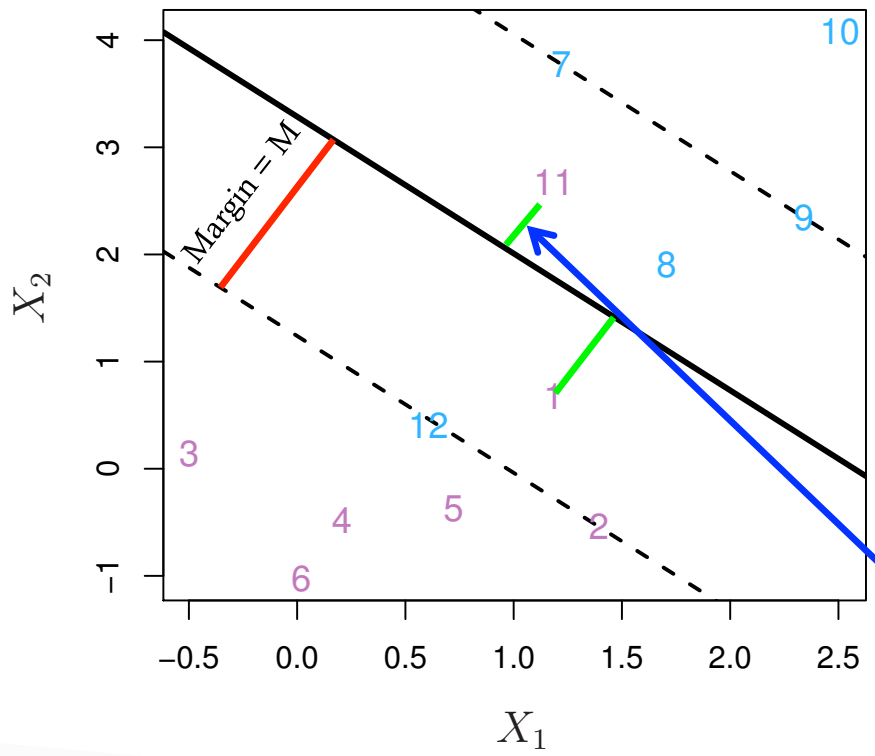
Slack Variables - ϵ_i

$\epsilon_i = 0 \Rightarrow$ Outside of Margin

$\epsilon_i \in (0, 1) \Rightarrow$ Inside of Margin

Support Vector Machines

Building Block #2: Support Vector Classifier



$$\text{Purple: } y_i = -1$$

$$\text{Blue: } y_i = +1$$

$$y_i(\beta_0 + \mathbf{x}_i'\boldsymbol{\beta}) \geq M(1 - \epsilon_i)$$

Slack Variables - ϵ_i

$\epsilon_i = 0 \Rightarrow$ Outside of Margin

$\epsilon_i \in (0, 1) \Rightarrow$ Inside of Margin

$\epsilon_i \geq 1 \Rightarrow$ On Wrong Side

Support Vector Machines

Building Block #2: Support Vector Classifier

$$y_i(\beta_0 + \mathbf{x}_i'\boldsymbol{\beta}) \geq M(1 - \epsilon_i)$$

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

Tuning Parameter - C

$C = 0 \Rightarrow$ Separating Hyperplane

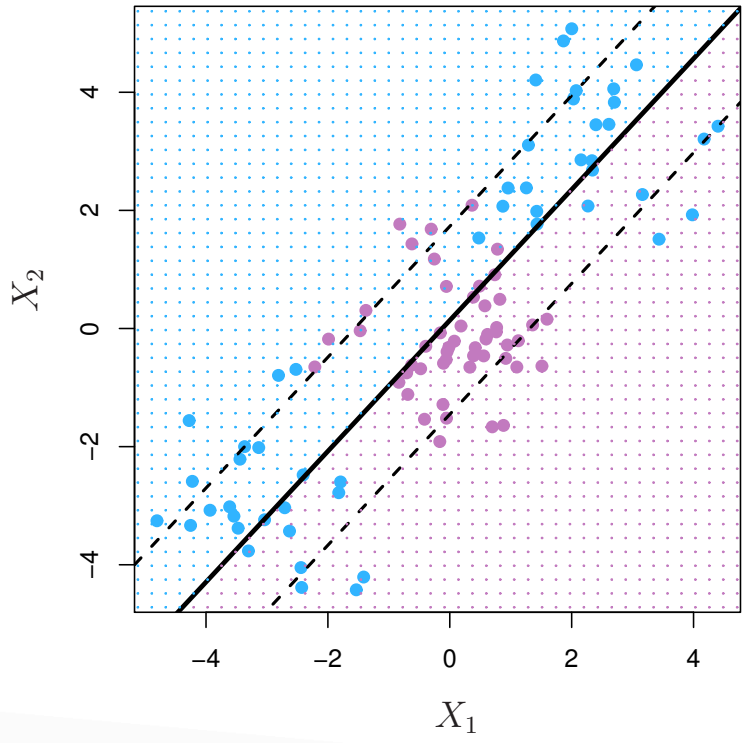
$C \uparrow \Rightarrow$ More Tolerant of Errors

For $C \geq 0$ no more than C obs. on wrong side of plane

Support Vector Machines

Support Vector Machines:

Big Idea: A plane, literally, just won't cut it! Need something non-linear.



Support Vector Machines

Support Vector Machines:

Big Realization: Hyperplane only depends on support vectors (its independent of everything else).

$$\beta_0 + \mathbf{x}_0'\boldsymbol{\beta} \equiv \beta_0 + \sum_{i \in \mathcal{S}} \mathbf{x}_0'(\beta_i^\star \mathbf{x}_i) = \beta_0 + \sum_{i \in \mathcal{S}} \beta_i^\star \mathbf{x}_0'\mathbf{x}_i$$

Big Idea: Re-write classifier function above using basis functions. That is,

$$\beta_0 + \sum_{i \in \mathcal{S}} \beta_i^\star B(\mathbf{x}_0, \mathbf{x}_i)$$

$\mathcal{S} =$ Support Set

$B =$ Basis (or Kernel) Function

Support Vector Machines

Support Vector Machines:

Common Choices of Kernels (Basis Functions):

$$\text{Polynomial}: B(\mathbf{x}_0, \mathbf{x}_i) = (1 + \mathbf{x}_0' \mathbf{x}_i)^d$$

$$\text{Radial}: B(\mathbf{x}_0, \mathbf{x}_i) = \exp\{-\gamma \|\mathbf{x}_0 - \mathbf{x}_i\|^2\}$$

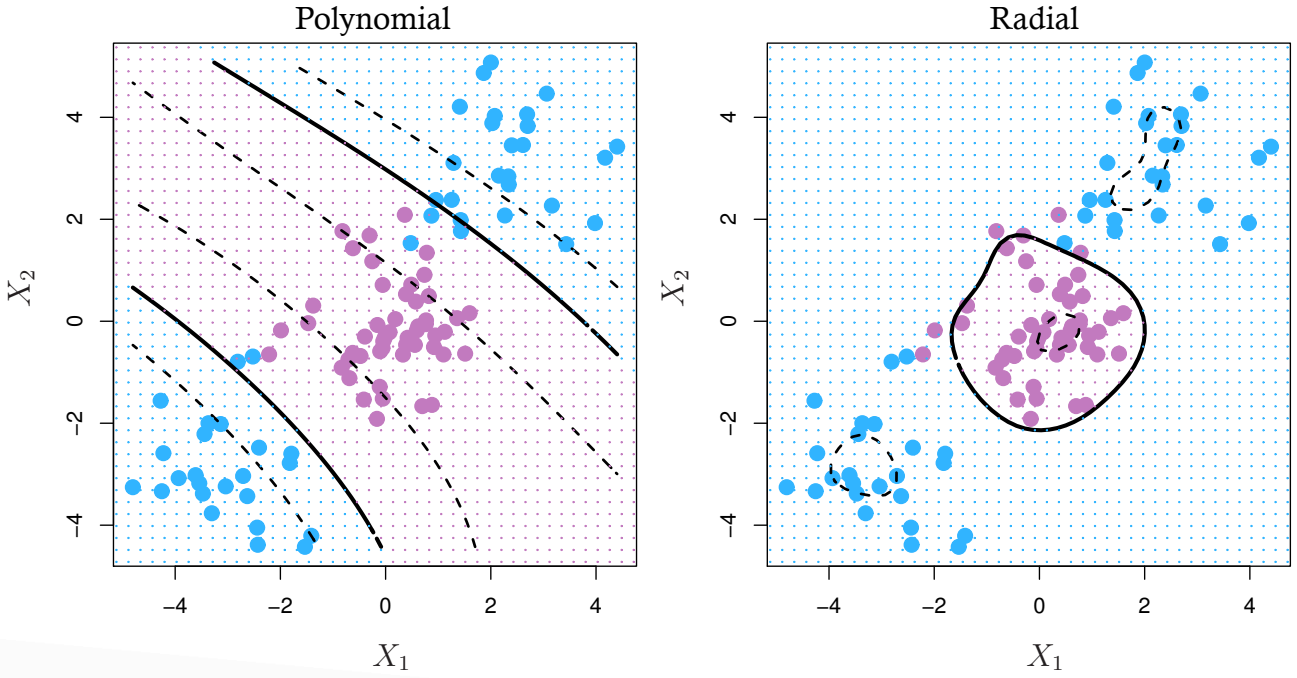
$$\text{Neural Network}: B(\mathbf{x}_0, \mathbf{x}_i) = \tanh\{\kappa_1 \mathbf{x}_0' \mathbf{x}_i + \kappa_2\}$$

Support Vector Machines

Support Vector Machines:

Issues:

1. Results change a lot depending on kernel (basis function).

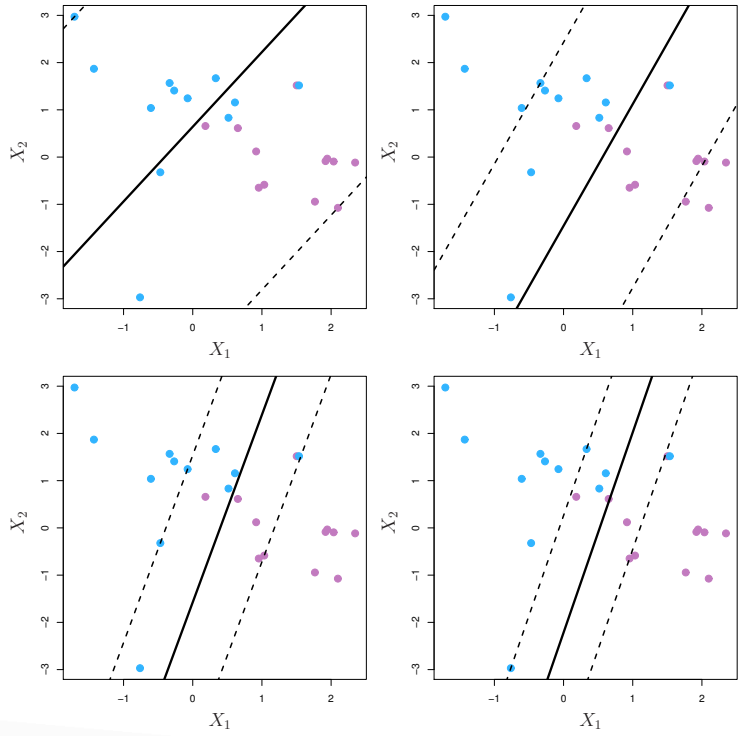


Support Vector Machines

Support Vector Machines:

Issues:

2. Tuning Parameter (C) chosen via cross-validation.



Classification Methods

Which method is appropriate for the car crashes data?

1. Logisitic Regression
2. Probit Regression
3. Discriminant Analysis
4. K-Nearest Neighbors
5. Support Vector Machines

It Depends

Classification Methods

Inference	Prediction
Logistic Regression	Discriminant Analysis
Probit Regression	K-Nearest Neighbors
	Support Vector Machines

Are you interested in inference or prediction?

Expectation for Car Crashes Analysis

1. Clean the data – you'll have to make decisions on
   - Should some categories be combined?
   - What do I do with the 99^{th} hour observations?
2. Determine an appropriate logistic/probit regression model for the car crashes dataset.
   - Justify which variables are in the model.
   - Justify linear vs. non-linear effects.
3. Fit your chosen model and report:
   - Effects and confidence intervals – be sure to interpret these appropriately.
   - Compare probabilities of different groups.
4. Results
   - One paragraph summary of what you did and why it was important.
   - Give some recommendations to the FHWA.

My Car Crashes Analysis

1. Clean the data – you'll have to make decisions on
   - Throw out 23 99th hour observations (only about 1%).
   - Fit a full model to see if categories have an issue.
   - Use VIF to check for collinearity
2. Determine an appropriate logistic/probit regression model for the car crashes dataset.
   - Forward selection using BIC – maybe even LASSO
   - Check few plots for non-linearity
3. Fit your chosen model and report:
   - Confidence intervals are easy – report on % change scale.
   - Compare probabilities of different groups (belt vs. no, helmet vs. no, Road.Type).
4. Results
   - One paragraph summary of what you did and why it was important.
   - Give some recommendations to the FHWA.