Introduction to Classification

-Introduction to Data Science-

ISL: Chapters 4.1, 4.2 ,4.3

Lecturer: Darren Homrighausen, PhD

An Overview of Classification

Some examples:

- A person arrives at an emergency room with a set of symptoms that could be 1 of 3 possible conditions. Which one is it?
- A online banking service must be able to determine whether each transaction is fraudulent or not, using a customer's location, past transaction history, etc.
- Given a set of individuals sequenced DNA, can we determine whether various mutations are associated with different phenotypes?

All of these problems are not regression problems. They are classification problems.

The Set-up

It begins just like regression: suppose we have observations

$$\mathcal{D} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

Again, we want to estimate a function that maps X into Y that helps us predict as yet observed data.

(This function is known as a classifier)

The same constraints apply:

- We want a classifier that predicts test data, not just the training data.
- Often, this comes with the introduction of some bias to get lower variance and better predictions.

Defining risk for classification

In regression, we have $Y_i \in \mathbb{R}$ and use squared error loss

$$\rightarrow \ell(f(X), Y) = (f(X) - Y)^2$$

Then

$$f_*(X) = \mathbb{E}[Y|X] = \operatorname*{argmin}_f \mathbb{E}(f(X) - Y)^2$$

is the Bayes' rule w.r.t. squared error loss

(Also known as the regression function and is the conditional expectation of Y at X)

What about when Y only takes on a few values?

Defining risk for classification

Instead, let $Y \in \mathcal{G}$, where $\mathcal{G}$ is a set of labels or classes

(Example: $\mathcal{G} = \{\text{threat}, \text{no threat}\}$)

Sometimes, the labels are encoded as integers

(Example: $\mathcal{G} = \{-1, 1\}$ or $\mathcal{G} = \{1, 2, \ldots, G\}$. The chosen integers are arbitrary)

As Y takes only a few values, 0-1 loss is natural

$$\ell(g(X), Y) = \mathbf{1}(Y \neq g(X)) \implies R(g) = \mathbb{E}[\ell(g(X), Y)] = ?$$

(? =)

Defining risk for classification

Instead, let $Y \in \mathcal{G}$, where $\mathcal{G}$ is a set of labels or classes

(Example: $\mathcal{G} = \{\text{threat}, \text{no threat}\}$)

Sometimes, the labels are encoded as integers

(Example: $\mathcal{G} = \{-1, 1\}$ or $\mathcal{G} = \{1, 2, \ldots, G\}$. The chosen integers are arbitrary)

As Y takes only a few values, 0-1 loss is natural

$$\ell(g(X), Y) = \mathbf{1}(Y \neq g(X)) \implies R(g) = \mathbb{E}[\ell(g(X), Y)] = ?$$

$(? = \mathbb{P}(g(X) \neq Y))$

Defining risk for classification

Instead, let $Y \in \mathcal{G}$, where $\mathcal{G}$ is a set of labels or classes

(Example: $\mathcal{G} = \{\text{threat}, \text{no threat}\}$)

Sometimes, the labels are encoded as integers

(Example: $\mathcal{G} = \{-1, 1\}$ or $\mathcal{G} = \{1, 2, \ldots, G\}$. The chosen integers are arbitrary)

As Y takes only a few values, 0-1 loss is natural

$$\ell(g(X), Y) = \mathbf{1}(Y \neq g(X)) \implies R(g) = \mathbb{E}[\ell(g(X), Y)] = ?$$

$(? = \mathbb{P}(g(X) \neq Y))$

Goal: Find a g such that $g(X) = Y$ as often as possible

Defining risk for classification

We have the Bayes' rule w.r.t. to 0-1 loss[1]:

$$
\begin{aligned}
g_*(X) &= \operatorname*{argmin}_{g \in \mathcal{G}} R(g) \\
&= \operatorname*{argmin}_{g \in \mathcal{G}} \mathbb{P}(Y \neq g | X) \\
&= \operatorname*{argmin}_{g \in \mathcal{G}} [1 - \mathbb{P}(Y = g | X)] \\
&= \operatorname*{arg\,max}_{g \in \mathcal{G}} \mathbb{P}(Y = g | X)
\end{aligned}
$$

(Interpretation: The Bayes rule for classification with this loss is to pick the class that maximizes the conditional probability of Y being that class)

[1] See section 2.4 in ESL for details

Example

In analogy to the regression function, the Bayes' rule for 0-1 loss looks like:

(If $\mathcal{G} = \{0, 1\}$)

$$g_*(X) = 0 \text{ if } \mathbb{P}(Y = 0|X) \geq \mathbb{P}(Y = 1|X)$$

or

$$g_*(X) = 1 \text{ if } \mathbb{P}(Y = 1|X) \geq \mathbb{P}(Y = 0|X)$$

(That is, we want to maximize the conditional probability)

Introductory example

An Introductory Example

Suppose we work for a credit card company and we wish to identify people that are likely to default on their credit card debt

We have features (for 10,000 people):

- Student status
- Income
- Balance

Along with their default status:

$$Y = \begin{cases} 1 \text{ if person defaults} \\ 0 \text{ if person doesn't default} \end{cases}$$

Let's look at some plots.

An Introductory Example

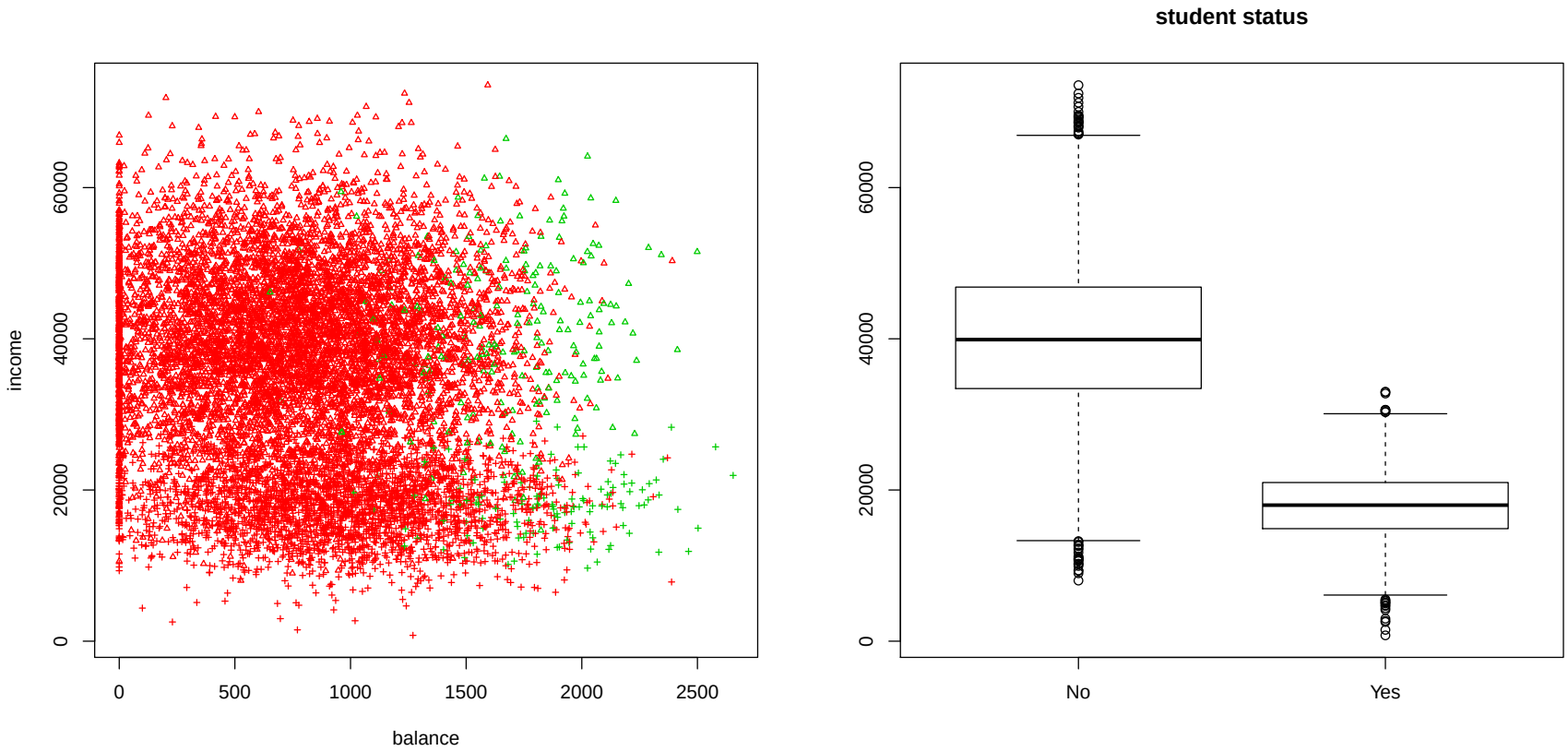
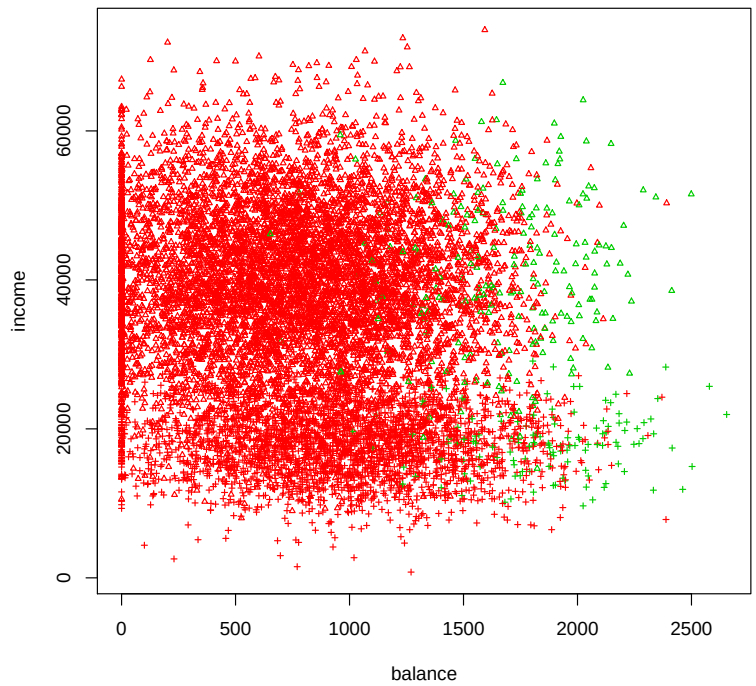
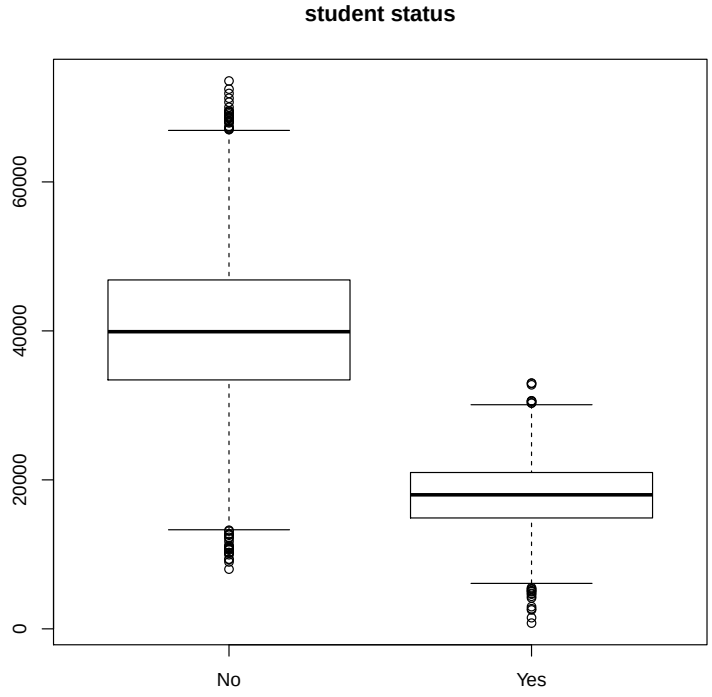


Figure: The red are people without defaults, green are defaults. The '+' are students, the 'Δ' are not students.

An Introductory Example





Some comments:

- Income doesn't seem to be related to defaults
- Student status is also unrelated to defaults, but highly related to income
- Balance seems to strongly predict default status.

An Introductory Example: Why not Use Regression?

Suppose for a moment we only consider balance. Then, we can run a simple linear regression of default status on balance

```
Y = rep(0,n)
Y[default == 'Yes'] = 1
out.lm = lm(Y~balance)
summary(out.lm)
```

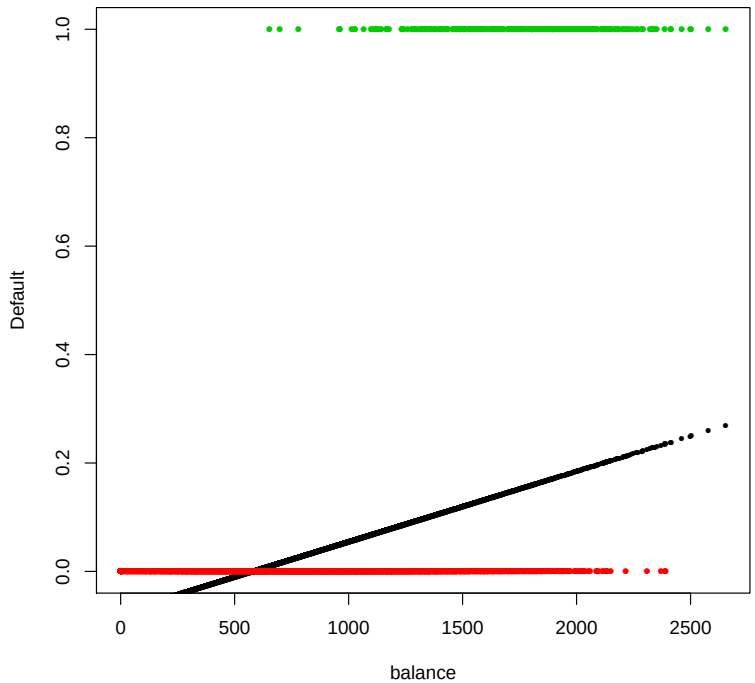
R will happily do this.

An Introductory Example: Why not Use Regression?

```
> summary(out.lm)
Call:
lm(formula = Y ~ balance)
Residuals:
     Min       1Q   Median       3Q      Max
-0.23533 -0.06939 -0.02628  0.02004  0.99046
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -7.519e-02  3.354e-03  -22.42   <2e-16 ***
balance      1.299e-04  3.475e-06   37.37   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
Residual standard error: 0.1681 on 9998 degrees of freedom
Multiple R-squared: 0.1226,Adjusted R-squared: 0.1225
F-statistic:  1397 on 1 and 9998 DF,  p-value: < 2.2e-16
```

An Introductory Example: Why not Use Regression?

Let's plot our data with estimated regression function:



Not so great..

Generalized Linear Models (GLMs)

GLMs generalize multiple regression via various distributions

Multiple regression:

$$Y_i = X_i^\top \beta + \epsilon_i$$

If instead, we model probabilities, then we are doing Logistic regression (with logit link):

Let $\pi(X_i) = \mathbb{P}(Y_i = 1|X_i)$,

$$\log\left(\frac{\pi(X_i)}{1-\pi(X_i)}\right) = X_i^\top \beta$$

(This is the logistic function)

It is differentiable, maps [0,1] to $\mathbb{R}$, and has inverse:

$$\pi(X_i) = \frac{\exp\{X_i^\top \beta\}}{1+\exp\{X_i^\top \beta\}}.$$

Generalized Linear Models (GLMs)

Let's look at each of these terms

- $\pi(X_i) = Pr(Y_i = 1|X_i)$ is the probability Y is equal to 1 at a given level of $X = X_i$
- $$\frac{\pi(X_i)}{1 - \pi(X_i)}$$

  is known as the odds that Y is equal to 1.
- $$\log\left(\frac{\pi(X_i)}{1 - \pi(X_i)}\right)$$

  is the log odds.

This models the log odds of $Y = 1$ as linear in the features X

Generalized Linear Models (GLMs)

With multiple regression, there is a closed form solution:

$$\hat{\beta} = (\mathbb{X}^\top \mathbb{X})^{-1} \mathbb{X}^\top Y$$

This was due to estimation via least squares

(This is also the maximum likelihood estimator (MLE) under Gaussian errors)

For GLMs, the likelihood is different, but we still use the MLE

There isn't any closed form solution and all solution methods are iterative maximizers of the likelihood

$$\arg\max_{\beta \in \mathbb{R}^p} \text{likelihood}(\beta) = \arg\max_{\beta \in \mathbb{R}^p} \prod_{i=1}^{n} \pi(X_i)^{Y_i} (1 - \pi(X_i))^{1-Y_i}$$

To do this, we can use standard implementations..

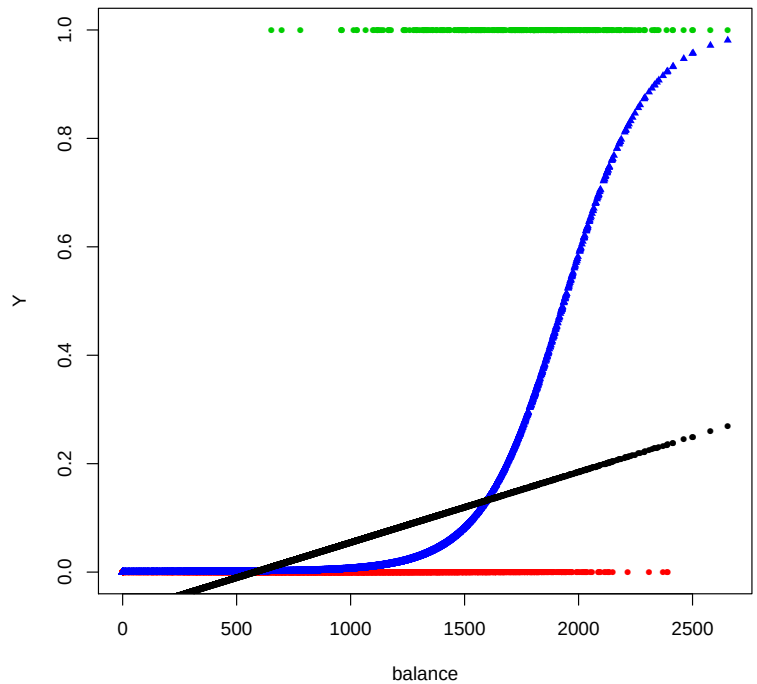
An Introductory Example: Generalized Linear Models (GLMs)

```
out.glm  = glm(default~balance,family='binomial')
> summary(out.glm)
Call:
glm(formula = default ~ balance, family = "binomial")
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2697  -0.1465  -0.0589  -0.0221   3.7589
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.065e+01  3.612e-01  -29.49   <2e-16 ***
balance      5.499e-03  2.204e-04   24.95   <2e-16 ***
---
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1596.5  on 9998  degrees of freedom
```

An Introductory Example: Compare GLM to Regression.

Let's plot our data with

- Simple Linear Regression (black)
- GLM (blue)



An Introductory Example: Making Predictions

Once we get $\hat{\beta}$, making predictions is a simple matter.

Suppose we want to estimate the probability that someone with a balance of \$1,000 will default. We form:

$$\hat{\pi}(1,000) = \frac{\exp\{-10.65 + 0.0055 * 1000\}}{1 + \exp\{-10.65 + 0.0055 * 1000\}} = 0.00576.$$

Pretty small..

Maybe look at \$2,000 instead:

$$\hat{\pi}(2,000) = \frac{\exp\{-10.65 + 0.0055 * 2000\}}{1 + \exp\{-10.65 + 0.0055 * 2000\}} = 0.586.$$

Much larger..

An Introductory Example: Classification

But, Darren, I thought we were classifying!

To form a classifier out of these predictions round the probabilities!

$$\hat{\pi}(1,000) = \frac{\exp\{-10.65 + 0.0055 * 1000\}}{1 + \exp\{-10.65 + 0.0055 * 1000\}} = 0.00576.$$

Thus, a balance of \$1,000 could be classified as no default

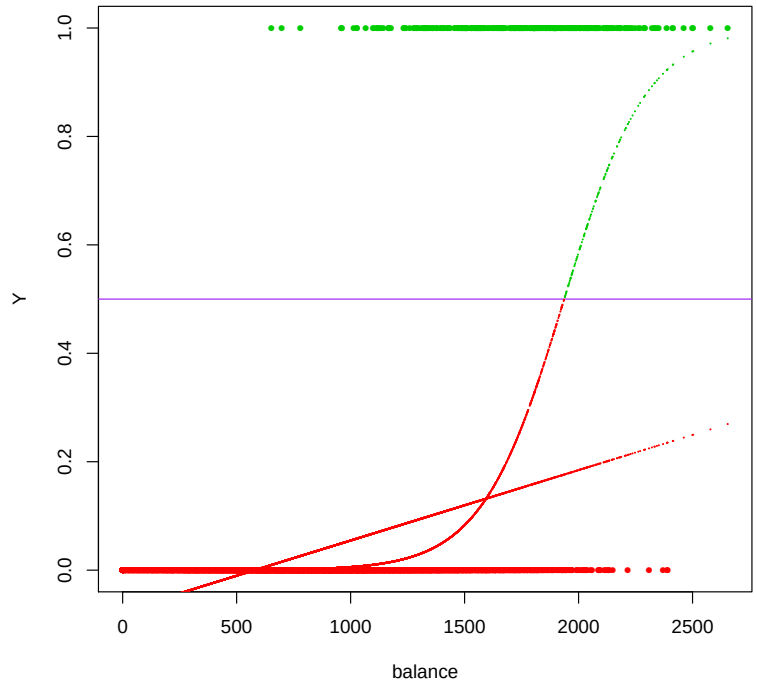
Maybe look at \$2,000 instead:

$$\hat{\pi}(2,000) = \frac{\exp\{-10.65 + 0.0055 * 2000\}}{1 + \exp\{-10.65 + 0.0055 * 2000\}} = 0.586.$$

A balance of \$2,000 could be classified as default

An Introductory Example: Compare GLM to Regression.

Results of using a cut-off of 0.5



(We discuss this cut-off point in more detail later)

Linear classifiers

Linear classifier

As our classifier $\hat{g}$ takes a discrete number of values, it is equivalent to partitioning the feature space into regions

The boundaries between these regions are known as decision boundaries

These decision boundaries are sets of points at which $\hat{g}$ is indifferent between classes

A linear classifier is a $\hat{g}$ that produces linear decision boundaries

Linear classifier: Example

Suppose $\mathcal{G} = \{-1, 1\}$ and we form the GLM logistic regression

This models the probabilities as

$$\mathbb{P}(Y = 1|X) = \frac{\exp\{\beta_0 + \beta^\top X\}}{1 + \exp\{\beta_0 + \beta^\top X\}}$$

$$\mathbb{P}(Y = -1|X) = \frac{1}{1 + \exp\{\beta_0 + \beta^\top X\}} = 1 - \mathbb{P}(Y = 1|X)$$

The logit transformation forms a linear decision boundary

$$\log\left(\frac{\mathbb{P}(Y = 1|X)}{\mathbb{P}(Y = -1|X)}\right) = \beta_0 + \beta^\top X$$

The decision boundary is the hyperplane $\{X : \beta_0 + \beta^\top X = 0\}$

We can form a classifier: Log-odds below 0, classify as -1, above 0 classify as a 1

(This is the same as rounding the probabilities)

The logistic elastic net

Logistic regression

Logistic regression maximizes the (log) likelihood:

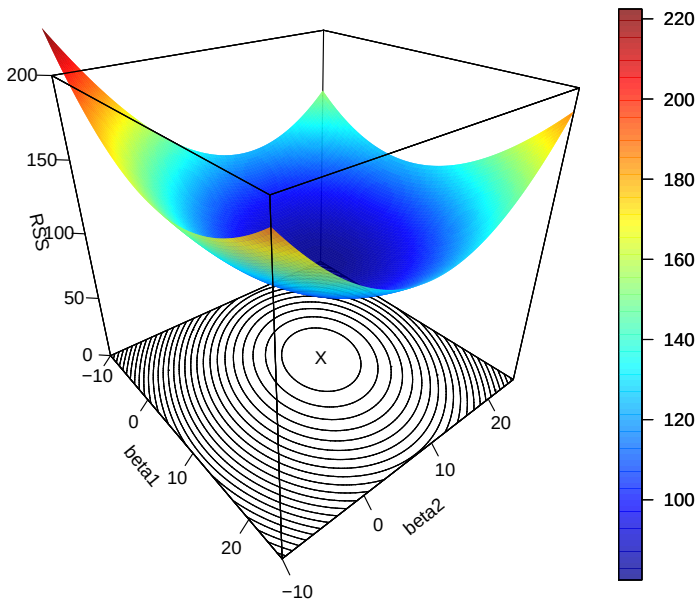
(Using $\pi_i(\beta) = \mathbb{P}(Y_i = 1|X_i, \beta)$ and $Y_i \in \{0, 1\}$)

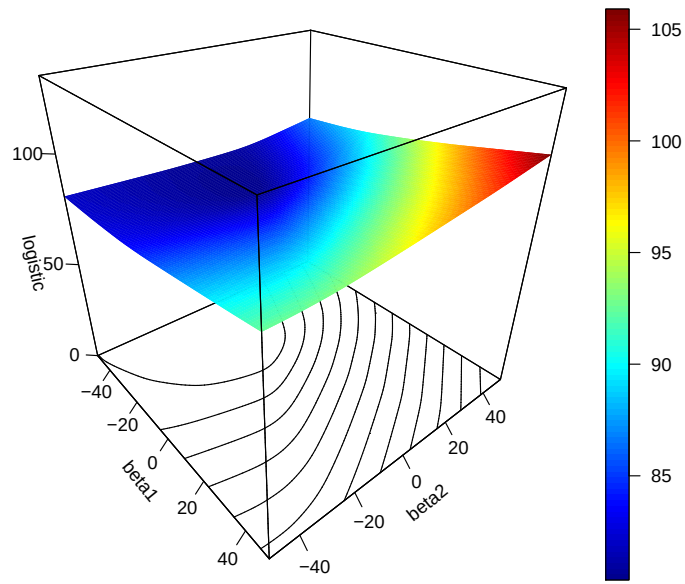
$$\begin{aligned}
\log \text{likelihood}(\beta) &= \log \prod_{i=1}^{n} \left(\pi_i(\beta)\right)^{Y_i} (1 - \pi_i(\beta))^{(1-Y_i)} \\
&= \sum_{i=1}^{n} \left(Y_i \log(\pi_i(\beta)) + (1 - Y_i) \log(1 - \pi_i(\beta))\right) \\
&= \sum_{i=1}^{n} \Big(Y_i \log(e^{\beta^\top X_i}/(1 + e^{\beta^\top X_i})) \\
&\qquad - (1 - Y_i) \log(1 + e^{\beta^\top X_i}) \Big) \\
&= \sum_{i=1}^{n} \left(Y_i \beta^\top X_i - \log(1 + e^{\beta^\top X_i}) \right)
\end{aligned}$$

If we write $\ell(\beta) = -\log \text{likelihood}(\beta)$, we have a loss function

Logistic regression

Different $\ell(\beta)$, will have different shapes:





$$\sum_{i=1}^{n} \left(Y_i - \beta^\top X_i\right)^2 \qquad -\sum_{i=1}^{n} \left(Y_i \beta^\top X_i - \log(1 + e^{\beta^\top X_i})\right)$$

Sparse logistic regression

This procedure suffers from all the problems of least squares

(We are minimizing the training error, after all)

We can use regularization techniques the same as before

Maximum likelihood for the Gaussian linear model:

$$\begin{aligned}
\arg\max_{\beta} \log(\text{likelihood}) &= \arg\max_{\beta} \log\left(\prod_{i=1}^{n} e^{-(Y_i - X_i^\top \beta)^2/2}\right) \\
&= \arg\max_{\beta} \sum_{i=1}^{n} -(Y_i - X_i^\top \beta)^2/2 \\
&\Leftrightarrow \operatorname{argmin}_{\beta} \sum_{i=1}^{n} (Y_i - X_i^\top \beta)^2 \\
&= \operatorname{argmin}_{\beta} ||Y_i - X_i^\top \beta||_2^2
\end{aligned}$$

We can do the same for logistic regression..

Sparse logistic regression

This means maximizing (over β_0, β):

$$\sum_{i=1}^{n} \Big(Y_i(\beta_0 + \beta^\top X_i) - \log(1 + e^{\beta_0 + \beta^\top X_i}) \Big) - \lambda(\alpha||\beta||_1 + (1-\alpha)||\beta||_2^2)$$

Or minimizing (over β_0, β):

$$\sum_{i=1}^{n} \Big(\log(1 + e^{\beta_0 + \beta^\top X_i}) - Y_i(\beta_0 + \beta^\top X_i) \Big) + \lambda(\alpha||\beta||_1 + (1-\alpha)||\beta||_2^2)$$

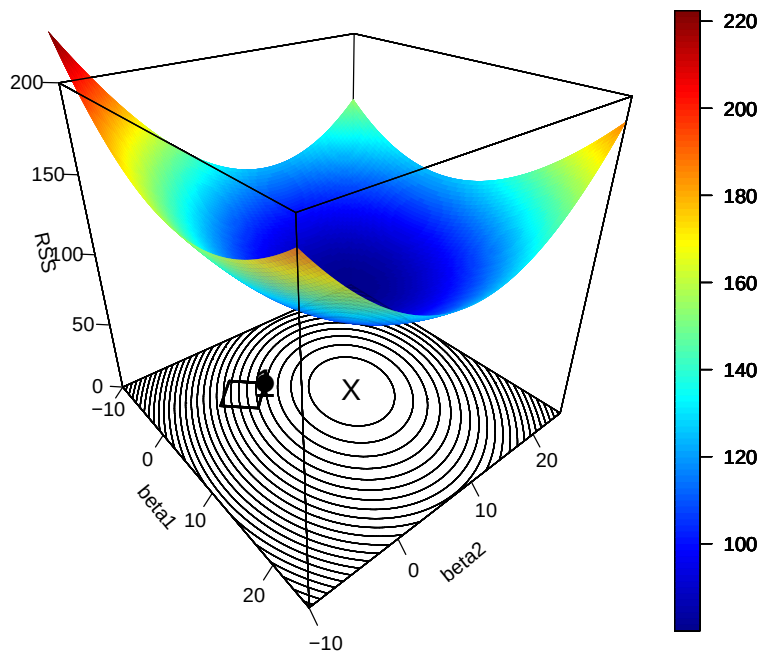
(Don't penalize the intercept and do standardize the features)

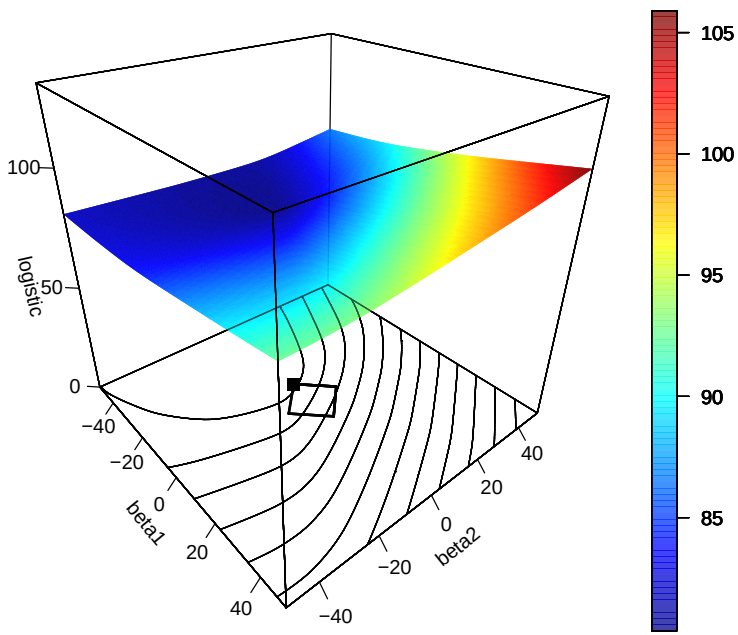
This is the logistic elastic net

Sparse logistic regression

$$\underset{\beta_0,\beta}{\operatorname{argmin}}\, \ell(\beta) + \lambda(\alpha||\beta||_1 + (1-\alpha)||\beta||_2^2)$$

(The pictures have $\alpha = 1$ and represent the constrained form)





$$\sum_{i=1}^{n} \left(Y_i - \beta^\top X_i\right)^2 \qquad -\sum_{i=1}^{n} \left(Y_i \beta^\top X_i - \log(1 + e^{\beta^\top X_i})\right)$$

Sparse logistic regression: Software

Using the R package glmnet finds the minimum CV solution over a grid of λ values for this new logistic loss function

Unfortunately, the computations are more difficult for path algorithms (such as the lars package) due to the coefficient profiles being only piecewise smooth

glmpath is an R package that does quadratic approximations to the profiles, while still computing the exact points at which the active set changes

(It is necessary to set a 'step' size argument for the approximation)

Sparse logistic regression: glmnet

Using glmnet we can do the following

```
glmnet(X,Y,family='binomial')
```

We can also do other likelihoods:

- Poisson
- multinomial
- Cox hazard
- Multivariate Gaussian

Other linear classifiers

We will cover other linear classifiers as well, including linear discriminant analysis (LDA) and support vector classifiers (SVC)...