Machine Learning & Data Mining

Logistic Regression

Kyung-Ah Sohn

Ajou University

Content

- Introduction
- Simple Logistic Regression
- Multiple logistic regression
- Training logistic regression

INTRODUCTION

Linear model

- Some algorithms are slow at runtime
  - e.g. K-NN

- Linear models are very fast, both to train and to use at runtime

- Simpler (e.g. linear) models are more interpretable

Example: Pass/Fail prediction

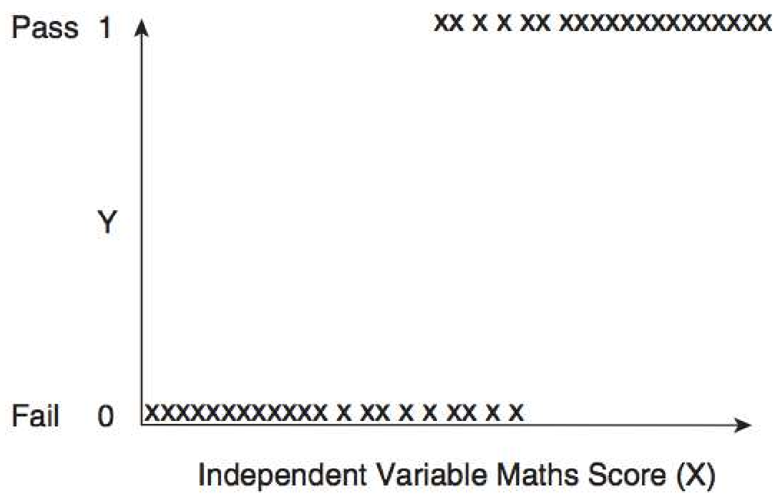
- We would like to predict whether a student will pass or fail an accountancy exam.

- The Y (pass?) variable is <u>categorical</u>: 0 or 1

- The X variable is a numerical value which specifies the student's math exam score.

- Can we use Linear Regression when Y is categorical?

Example (single explanatory variable)



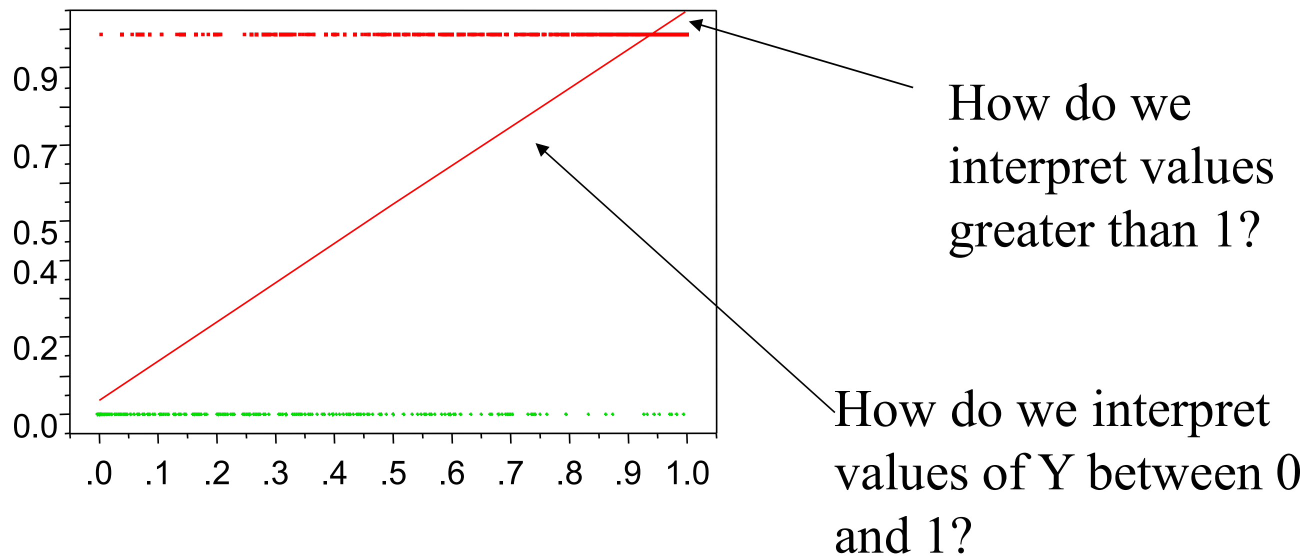
- x: math exam score
- y: pass or fail on the accountancy exam

$$y = \beta_0 + \beta_1 x?$$

http://www.uk.sagepub.com/burns/website%20material/Chapter%2024%20-%20Logistic%20regression.pdf

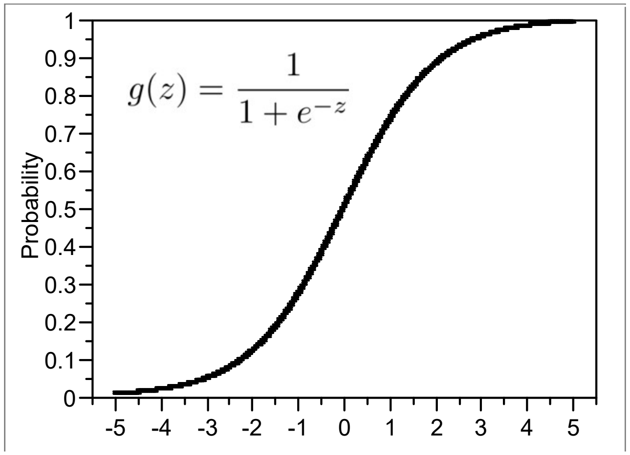
Why not Linear Regression?

- When Y only takes on values of 0 and 1, why standard linear regression is inappropriate?



Solution: Use Logistic Function

- Instead of trying to predict Y, let's try to predict **P(Y = 1)**, i.e., the probability a student will pass the exam
- Thus, we can model P(Y = 1) using a function that gives outputs between 0 and 1.
- We can use the logistic function
- Logistic Regression!



Logistic regression

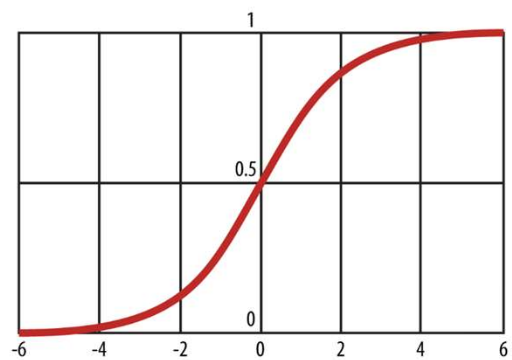
- The output of a logistic regression model is the probability of each class
- You can use these probabilities directly, or you could find a threshold so that you can predict either 1 or 0
- Unlike with linear regression – which predicts the actual value- the aim of logistic regression isn't to predict the actual value (0 or 1), but to output a probability.

Underlying Math

- You want a function that takes the data and outputs a value between 0 ~ 1.

$$P(t) = \text{logit}^{-1}(t) \equiv \frac{1}{\left(1+e^{-t}\right)} = \frac{e^t}{1+e^t}$$

Sigmoid function



- When t is very large, the value is close to 1.
- When t is very small, the value is close to 0.

SIMPLE LOGISTIC REGRESSION

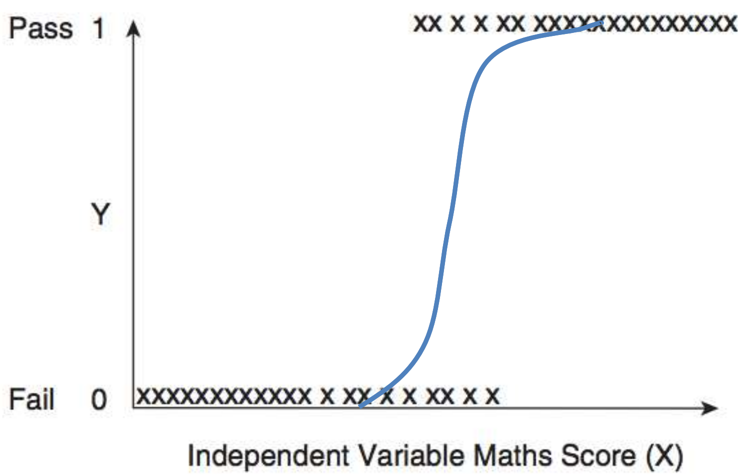
Simple Logistic Regression

- Logistic regression is very similar to linear regression

$$p = P(Y = 1|X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

- We have similar problems and questions as in linear regression
  - e.g. Is β_1 equal to 0? How sure are we about our guesses for β_0 and β_1?

Probability of success (y=1)

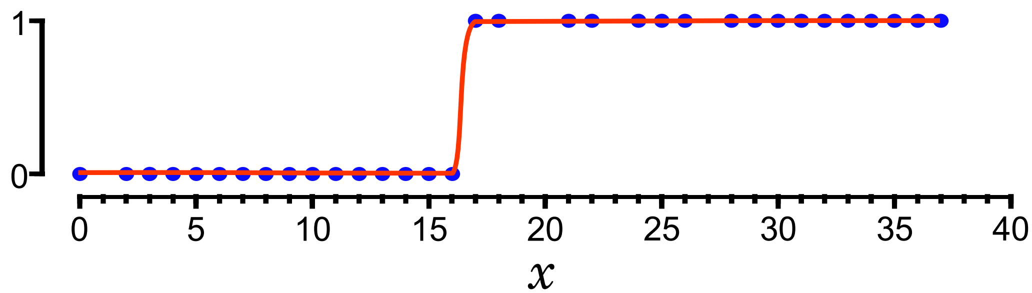


- The outcome is a probability of belonging to one of two conditions of Y, which can take any value between 0 and 1
  (rather than just 0 or 1)

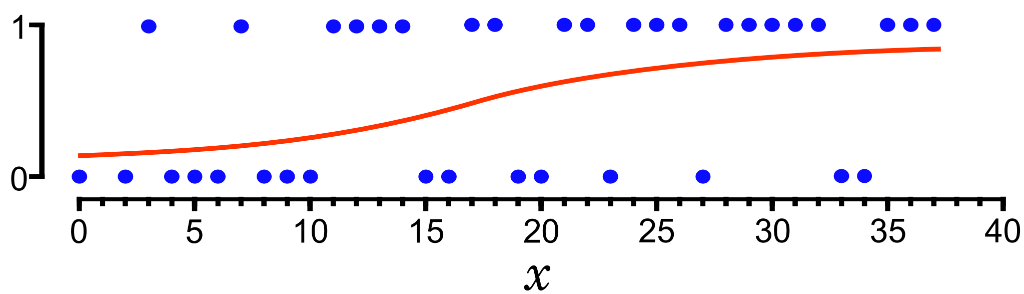
- $p(y=1|x=82)$: probability of passing the exam when the math score was 82

We wish to choose the best curve to fit the data.

Data that has a sharp survival cut off point between two classes (0 or 1) should have a large value of β_1.



Data with a lengthy transition from 0 to 1 should have a low value of β_1.



Interpreting β_1

- Interpreting what β_1 means is not very easy with logistic regression, simply because we are predicting P(Y) and not Y.
- If $\beta_1 = 0$, this means that there is no relationship between Y and X.
- If $\beta_1 > 0$, this means that when X gets larger so does the probability that Y = 1.
- If $\beta_1 < 0$, this means that when X gets larger, the probability that Y = 1 gets smaller.
- But how much bigger or smaller depends on where we are on the slope

odds

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

- After a bit of manipulation:

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}$$

odds

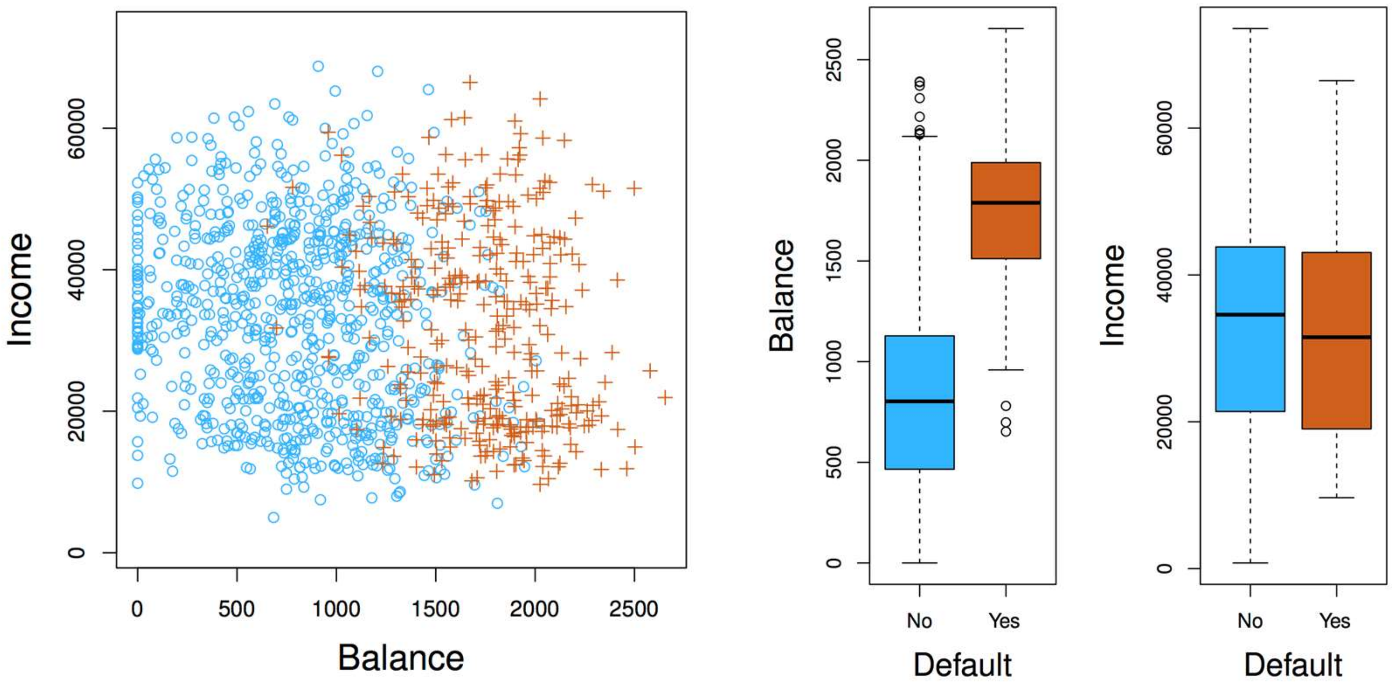
  - e.g. 1 in 5 people with an odds of ¼ will default
  - Traditionally used instead of probabilities in horse-racing
- Log-odds (logit) is linear in logistic regression

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

Example: Credit Card Default Data

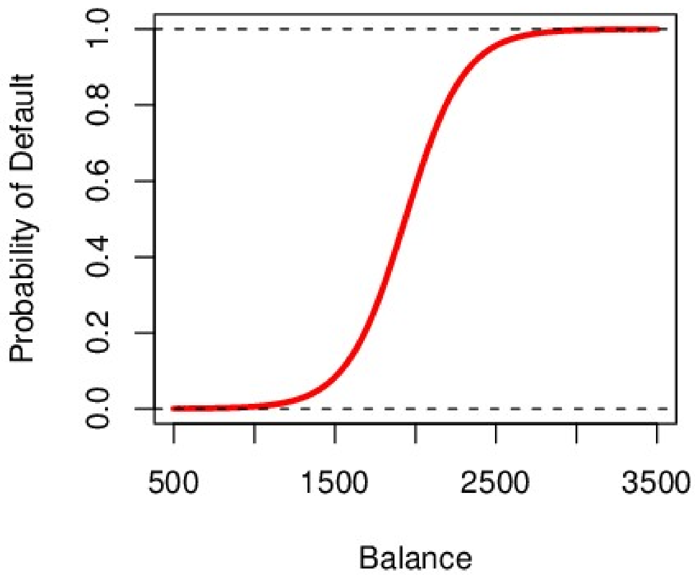
- We would like to be able to predict customers that are likely to default
- Possible X variables are:
  - Annual Income
  - Monthly credit card balance
- The Y variable (Default) is <u>categorical</u>: Yes or No
- How do we check the relationship between Y and X?

The Default Dataset



Logistic Function on Default Data

- Now the probability of default is close to, but not less than zero for low balances. And close to but not above 1 for high balances



Are the coefficients significant?

- We still want to perform a hypothesis test to see whether we can be sure that are β_0 and β_1 significantly different from zero.
- Here the p-value for balance is very small, and β_1 is positive, so we are sure that if the balance increase, then the probability of default will increase as well.

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.6513	0.3612	-29.5	< 0.0001
balance	0.0055	0.0002	24.9	< 0.0001

Making Prediction

- Suppose an individual has an average balance of $1000. What is their probability of default?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.00576$$

- The predicted probability of default for an individual with a balance of $1000 is less than 1%.

- For a balance of $2000, the probability is much higher, and equals to 0.586 (58.6%).

MULTIPLE LOGISTIC REGRESSION

Multiple Logistic Regression

- Multiple variables case
- We can fit multiple logistic just like regular regression

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}.$$

Multiple Logistic Regression
- Default Data -

- Predict Default using:
  - Balance (quantitative), Income (quantitative), Student (qualitative)

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	-0.6468	0.2362	-2.74	0.0062

- Predictions: A student with a credit card balance of $1,500 and an income of $40,000 has an estimated probability of default

$$\hat{p}(X) = \frac{e^{-10.869+0.00574\times 1500+0.003\times 40-0.6468\times 1}}{1+e^{-10.869+0.00574\times 1500+0.003\times 40-0.6468\times 1}} = 0.058.$$

An Apparent Contradiction!

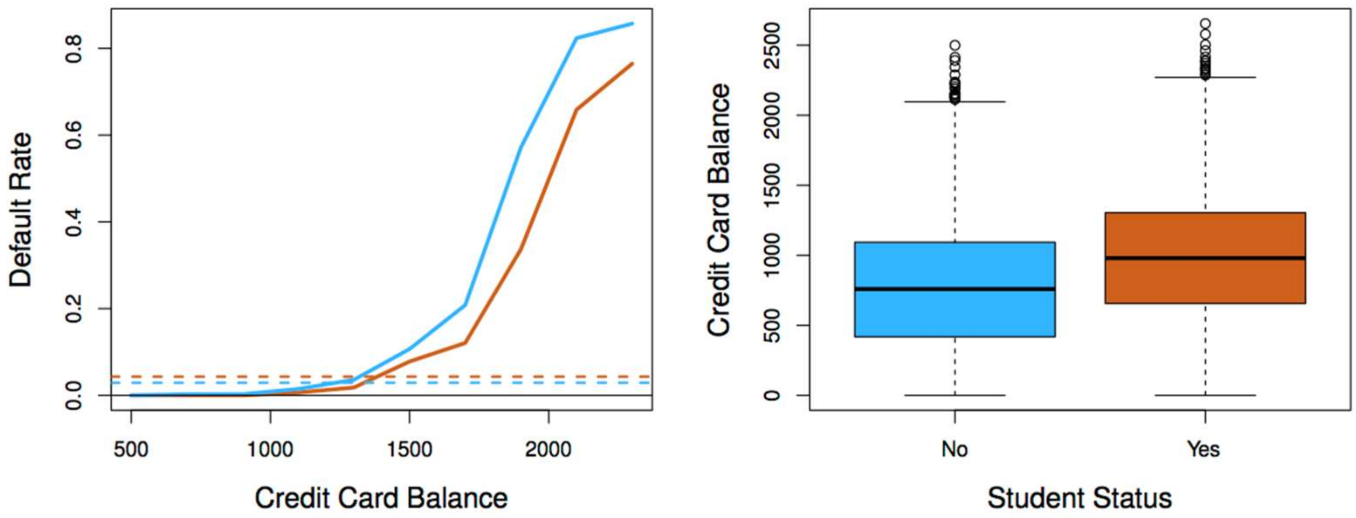
	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

Positive

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	-0.6468	0.2362	-2.74	0.0062

Negative

Students (Orange) vs. Non-students (Blue)



To whom should credit be offered?

- A student is risker than non students if no information about the credit card balance is available

- However, that student is less risky than a non student with the same credit card balance!

Decision boundary of logistic regression model

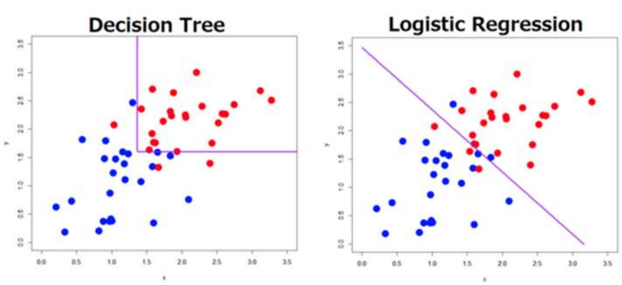
Where P(Y=1| X) == P(Y=0| X)?

$$\frac{\exp(\beta_0+\beta_1 x_1+\ldots+\beta_p x_p)}{1+\exp(\beta_0+\beta_1 x_1+\ldots+\beta_p x_p)}=\frac{1}{1+\exp(\beta_0+\beta_1 x_1+\ldots+\beta_p x_p)}$$

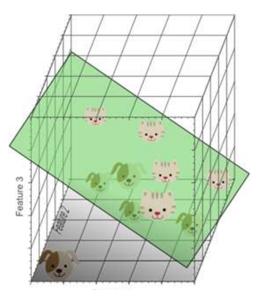
$$\exp(\beta_0+\beta_1 x_1+\ldots+\beta_p x_p)=1$$

$$\beta_0+\beta_1 x_1+\ldots+\beta_p x_p=0$$

: Linear classifier



- 1D – threshold
- 2D – linear line
- 3D – plane



Multi-class classification

- Multinomial logistic regression (softmax regression)

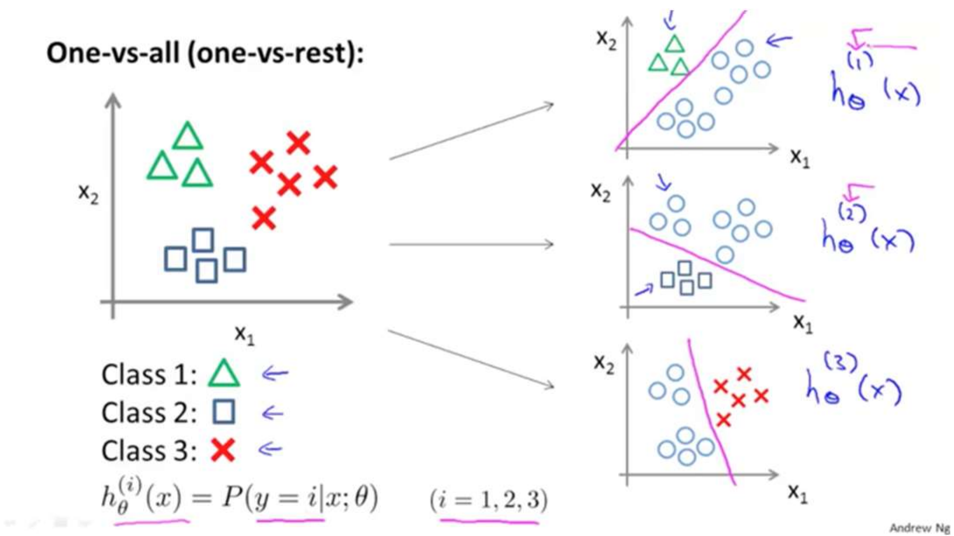
$$p(y=k|x) = \frac{\exp w_k^\top x}{\sum_j \exp w_j^\top x}$$

Softmax function

- Recommended for *mutually exclusive* classes (each sample can only belong to a single class)

Multi-class classification

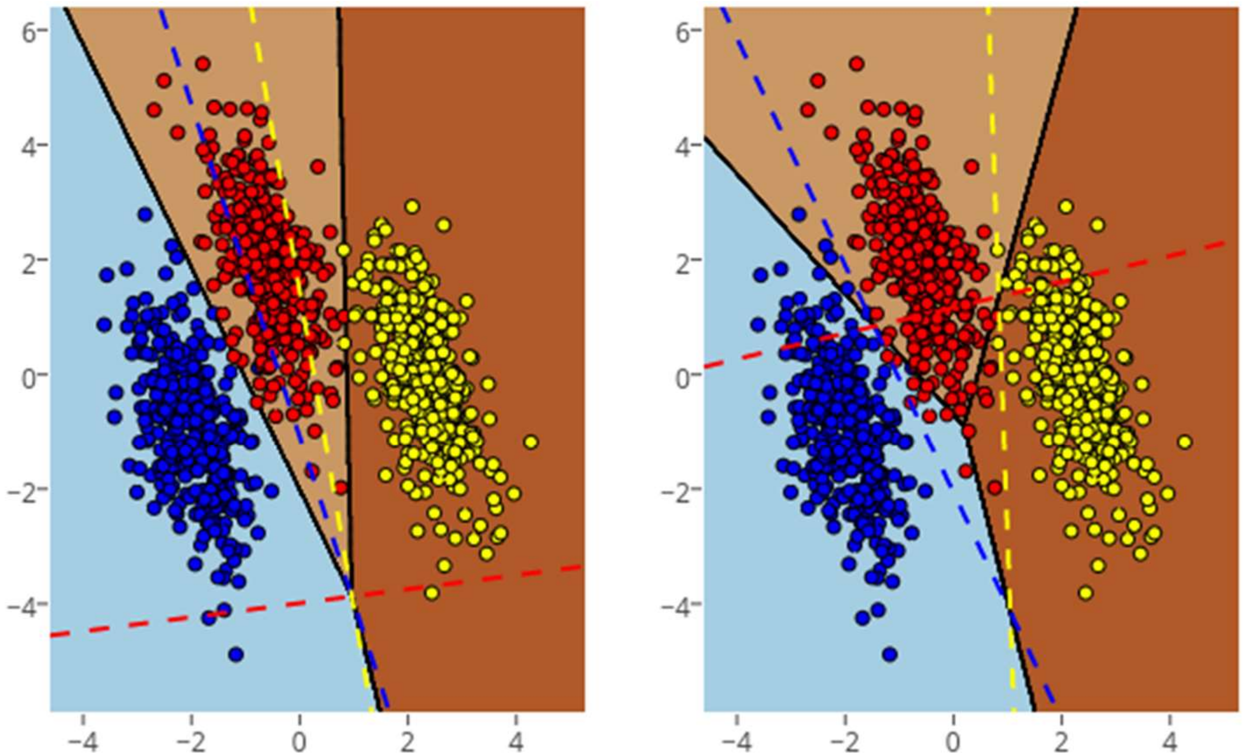
- Or, One versus Rest (OvR, or OvA: one vs. all)
  - Train binary logistic regression classifier for each class k to predict probability of y=k
  - On new x, predict class k which has the maximum probability value



Decision boundary

Multinomial

One-versus-rest



TRAINING LOGISTIC REGRESSION

Training a logistic regression model

- More complex than the case of linear regression
- Need to optimize β so that the model gives the best possible reproduction of training set labels
  - Usually done by numerical approximation of maximum likelihood estimation (MLE)
  - On really large datasets, may use stochastic gradient descent

Cost function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\hat{y}_i = \begin{cases} 1 & if\ \sigma(z_i) \geq 0.5 \\ 0 & otherwise \end{cases}$$

$$z = w_0 + w_1 x_1 + \cdots + w_P x_p$$

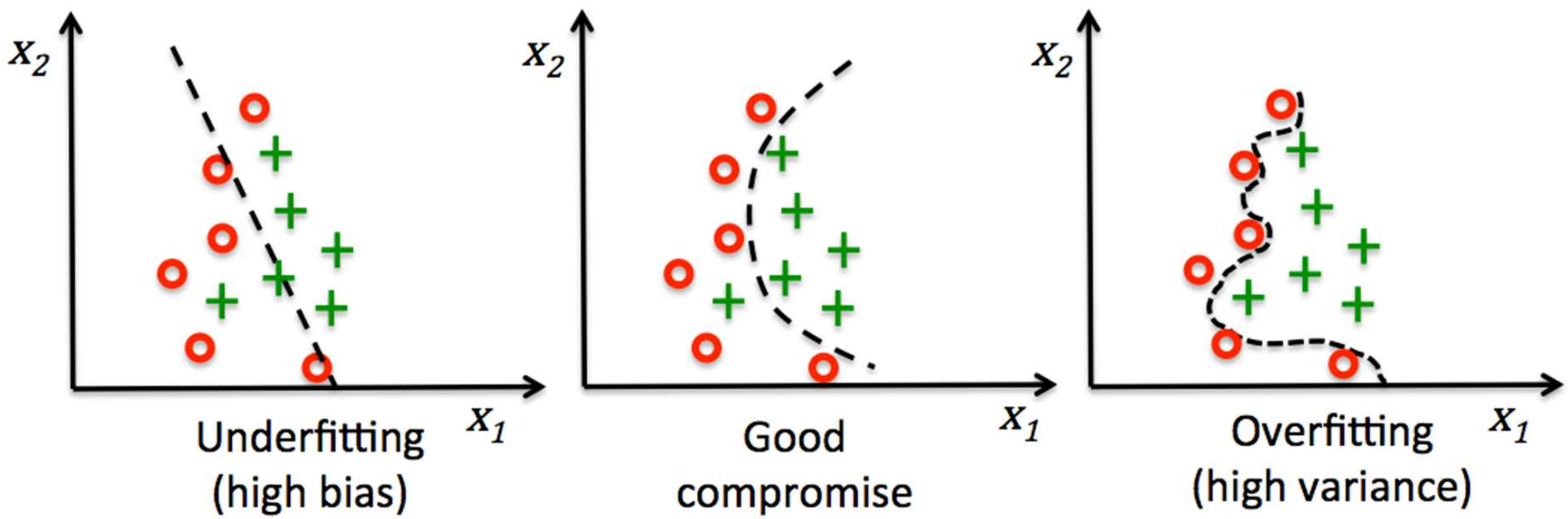
Likelihood $\quad L(w) = p(y|x; w) = \prod_i^n (\sigma(z_i))^{y_i} (1 - \sigma(z_i))^{1-y_i}$

Log-likelihood $\quad l(w) = \log L(w) \ = \sum_i^n y_i \log\big(\sigma(z_i)\big) + (1 - y_i) \log\big(1 - \sigma(z_i)\big)$

Negative Log-likelihood $\quad J(w) = -l(w)$

Minimize the cost function using gradient descent

Overfitting



Regularized logistic regression

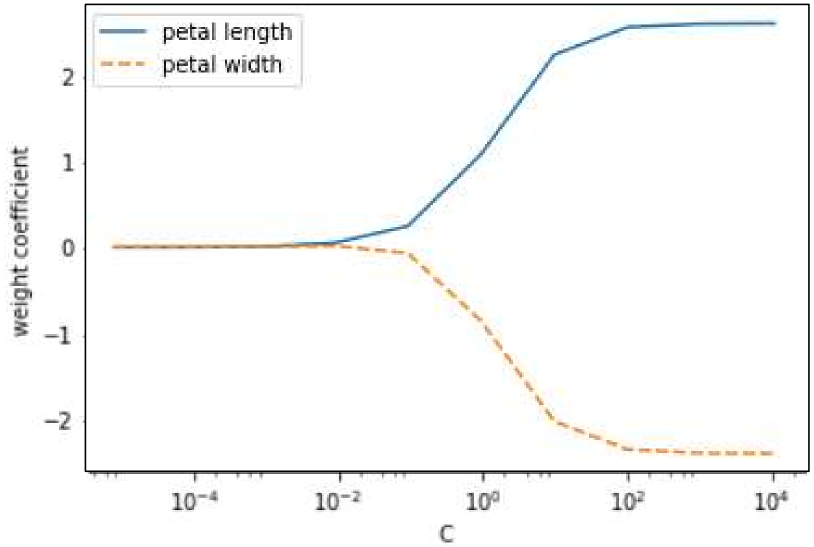
- Tune the model complexity via regularization
- e.g. Add L2 penalty on the cost function

$$J(w) = \sum_{i}^{n} \left[-y_i \log\big(\sigma(z_i)\big) - (1 - y_i) \log\big(1 - \sigma(z_i)\big) \right] + \frac{\lambda}{2} ||w||^2$$

Coefficient shrink

- Weight coefficients shrink if we decrease parameter C, that is, if we increase the regularization strength

```
weights, params = [], []
for c in np.arange(-5., 5.):
        lr = LogisticRegression(C=10.**c, random_state=0)
        lr.fit(X_train_std, y_train)
        weights.append(lr.coef_[1])
        params.append(10**c)
```



Logistic regression: summary

- Advantages
  - Makes no assumptions about distributions of classes in feature space
  - Easily extended to multiple classes
  - Quick to train, fast at classifying new data
  - Good accuracy for many simple data sets
  - Can interpret model coefficients as indicators of feature importance
- Disadvantages
  - Linear decision boundary

Summary: regression models

- Regression models can be used to describe the average effect of predictors on outcomes in your data set.
- They can look at each predictor "adjusting for" the others (estimating what would happen if all others were held constant.)
- Removing redundant predictors (variable selection) is key to achieving predictive accuracy and robustness