# 2108-351-351m-451-w15a-f-Classification-Supervised Learning

Profs: R. H. French, L. S. Bruckman, P. Leu, K. Davis, S. Cirlos

TAs: W. Oltjen, K. Hernandez, M. Li, M. Li, D. Colvin

30 November, 2022

## Contents



#### 15.1.1.1 Class Readings, Assignments, Syllabus Topics

##### 15.1.1.1.1 Course Evaluations Are Open Now

- lets get to 90% response rate
- We want statistically significant results!
  - I look for suggestions on how to improve the course
- https://webapps.case.edu/courseevals/

#### 15.1.1.2 Machine Learning Portal

- A very useful resource

Also another useful overview of Machine Learning

- Is in 3-readings/3-CheatSheets
  - super-cheatsheet-machine-learning.pdf
- And the two cheat sheets in 3-readings/3-CheatSheets
  - supervised-learning.pdf
  - unsupervised-learning.pdf

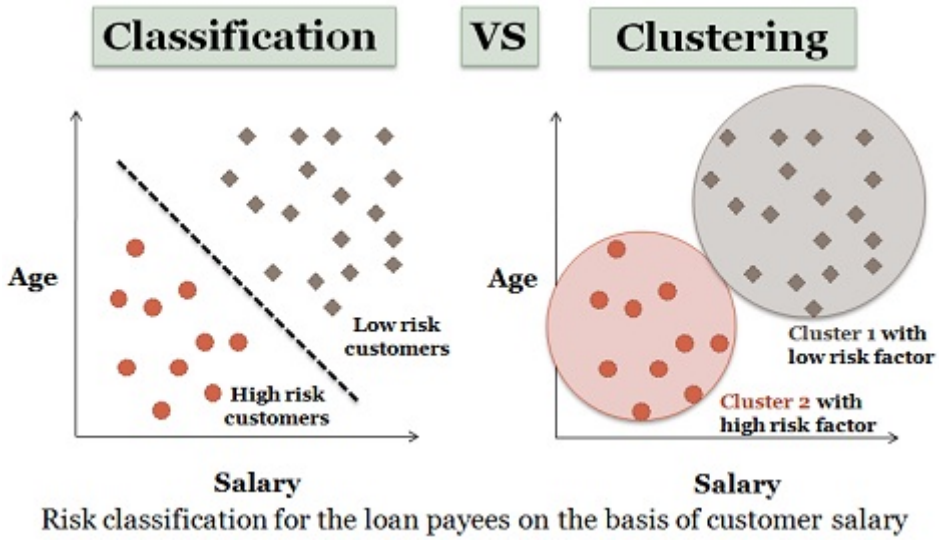



Figure 1: Classification vs. Clustering

#### 15.1.1.3 Classification vs. Clustering; Whats the difference

- Classification and Clustering are the two types of learning methods
  - which characterize objects into groups by one or more features.

These processes appear to be similar,

- but there is a difference between them
  - in context of **data mining**.

The prior difference between classification and clustering

- is that classification is used in **supervised learning** techniques
  - where predefined labels are assigned to instances by properties,
- on the contrary, clustering is used in **unsupervised learning**
  - where similar instances are grouped,
  - based on their features or properties

When the training is provided to the system,

- the class label of training "tuple" is known and then tested,
  - (What is a "tuple"
    - a tuple is a finite ordered list (sequence) of elements)
- this is known as supervised learning.

On the other hand, unsupervised learning

- does not involve training or learning,
- and the training sample is not known previously.

#### 15.1.1.3.1 Definition of Classification

- Classification is the process of learning a model
  - that elucidate different predetermined classes of data.
  - It is a two-step process,
    - comprised of a learning step and a classification step.
  - In the learning step, a classification model is constructed
    - and in the classification step the constructed model is used
    - to prefigure the class labels for given data.

For example, in a banking application,

- the customer who applies for a loan
  - may be classified as a safe and risky
  - according to his/her age and salary.
- This type of activity is also called supervised learning.
  - The constructed model can be used to classify new data.
- The learning step can be accomplished
  - by using already defined training set of data.
- Each record in the training data
  - is associated with an attribute referred to as a class label,
  - that signifies which class the record belongs to.
- The produced model could be in the form
  - of a decision tree or in a set of rules.

A decision tree is a graphical depiction

- of the interpretation of each class or classification rules.
- Regression is the special application of classification rules.
- Regression is useful when
  - the value of a variable
  - is predicted based on the tuple
  - rather than mapping a tuple of data
  - from a relation to a definite class.
- Some common classification algorithms are
  - decision tree,
  - neural networks,
  - logistic regression, etc.

#### 15.1.1.3.2 Definition of Clustering

- Clustering is a technique of organising a group of data
  - into classes and clusters
- where the objects reside inside a cluster
  - will have high similarity
- and the objects of two clusters
  - would be dissimilar to each other.
- Here the two clusters can be considered as disjoint.
- The main target of clustering
  - is to divide the whole data into multiple clusters.
- Unlike classification process,
  - here the class labels of objects are not known before,
  - and clustering pertains to unsupervised learning.

In clustering,

- the similarity between two objects

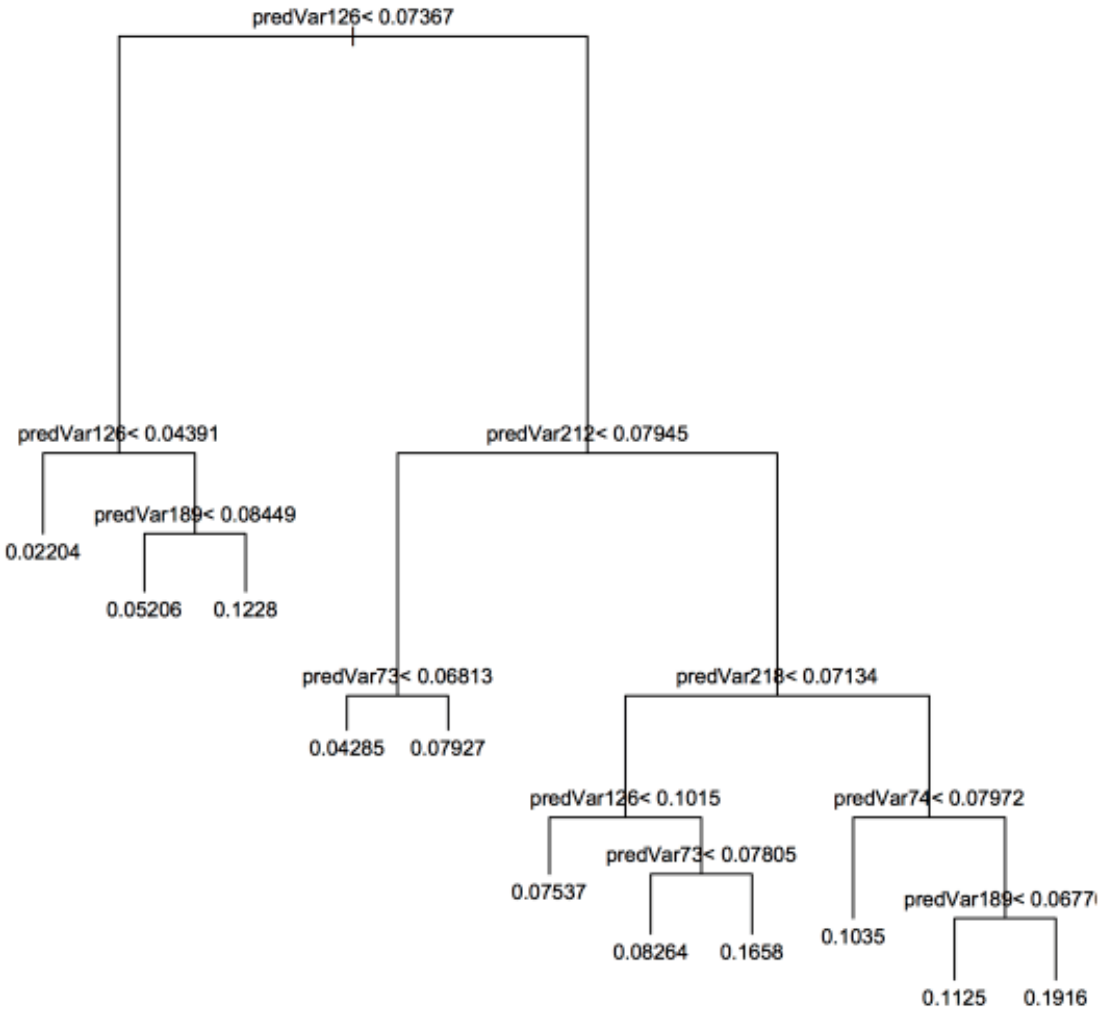



Figure 2: Decision Tree

  - is measured by the similarity function
  - where the distance between those two object is measured.
- Shorter the distance higher the similarity,
  - conversely longer the distance higher the dissimilarity.

Another example of clustering,

- there are two clusters named as mammal and reptile.
- A mammal cluster includes
  - human,
  - leopards,
  - elephant,
  - etc.
- On the other hand, reptile cluster includes
  - snakes,
  - lizard,
  - komodo dragon etc.
- The tools mainly used in cluster analysis are
  - k-means,
  - k-medoids,
  - density based,
  - hierarchical
  - and several other methods.

#### 15.1.1.3.3 Key Differences Between Classification and Clustering

- Classification is the process of classifying the data
  - with the help of class labels.

On the other hand, Clustering is similar to classification

- but there are no predefined class labels.

Classification is supervised learning.

- While clustering is also known as unsupervised learning.

A training sample is provided in a classification method

- while in case of clustering training data is not provided.

### 15.1.1.4 Overview of Clustering

- Cluster analysis or clustering is the task
  - of grouping a set of objects
    - in such a way that objects in the same group (called a cluster)
    - are more similar (in some sense) to each other
    - than to those in other groups (clusters).

It is a main task of exploratory data mining,

- and a common technique for statistical data analysis,
- used in many fields, including
  - machine learning,
  - pattern recognition,
  - image analysis,
  - information retrieval,
  - bioinformatics,
  - data compression,

  - and computer graphics.

#### 15.1.1.4.1 K-means Clustering

- is a method of vector quantization,
  - originally from signal processing,
  - that is popular for cluster analysis in data mining.

k-means clustering aims to partition

- n observations into k clusters
  - in which each observation belongs to the cluster with the nearest mean,
  - serving as a prototype of the cluster.
- This results in a partitioning of the data space into Voronoi cells.

#### 15.1.1.4.2 Hierarchical Clustering Agglomerative Hierarchical Clustering

- This is a "bottom up" approach:
  - each observation starts in its own cluster,
  - and pairs of clusters are merged as one moves up the hierarchy.

Divisive Hierarchical Clustering

- This is a "top down" approach:
  - all observations start in one cluster,
  - and splits are performed recursively as one moves down the hierarchy.

---

### 15.1.1.5 Classification (Supervised Learning)

- Supervised learning example
  - Classifying things into two cateogories
    - eye color {brown, blue, green}
    - email {spam, ham}.
  - A categorical variable gives labels to the objects

Can we use linear regression for classification problems?

For binary classification, linear regression does a decent job.

This is called linear discriminant analysis

Conditional mean of $Y$ given $X = x$.

- $E(Y|X = x) = Pr(Y = 1|X = x)$

Linear regression can produce probabilities less than 0 or greater than 0

Instead Logistic Regression is more appropriate.

Cateogorical problems.

- Linear regression is not appropriate.
- Multi-class logistic regression is better.

Similar to encoding levels of cateogorical variables

- into a series of bits
- that each have only two levels.

#### 15.1.1.5.1 Logistic Regression

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

Monotone transformation gives us a logarithmic ln function

- $log(p(X)/(1 - p(X))) = \beta_0 + \beta_1(X)$

[note in R, *log* is the natural log *ln*]

This is the "log odds" or the logit transformation of $p(X)$

Maximum Likelihood (Ronald Fisher)

Use Maximum Likelihood to estimate the parameters of the Logistic Regression model.

Using the glm package, as opposed to the lm package.

## Logistic Regression

Let's write $p(X) = \Pr(Y = 1|X)$ for short and consider using `balance` to predict `default`. Logistic regression uses the form

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

($e \approx 2.71828$ is a mathematical constant [Euler's number.])
It is easy to see that no matter what values $\beta_0$, $\beta_1$ or $X$ take, $p(X)$ will have values between 0 and 1.

A bit of rearrangement gives

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

This monotone transformation is called the *log odds* or *logit* transformation of $p(X)$.

And if you have multiple predictors and 1 categorical response,

- you can do multiple logistic regression,
- as a simple extension.

#### 15.1.1.5.2 Case Control Sampling

- If your data science study is to learn about
  - The rate of occurrence
  - Versus the benefits of a treatment

# Logistic regression with several variables

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

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

Figure 3: multiple logistic regression

You build your study population differently

- For treatment studies
- You use case control sampling
- So you have enough examples of the sickness

In epidemiology, you always want to use the cases of the disease, while then sampling from your control group.

The prevalence of disease in your study group (your sample) may be larger than in the population at large

So the probabilty of disease in you study sample

- (as opposed to the true population you pulled your sample from )
- might mean your logistic regression model is wrong.

Instead it turns out that only the $\beta_0$ term,

- the intercept will be wrong,
- the slopes ($\beta_1$ term) will be right.

So you can correct the slope to represent the actual prevalence in your real population.

- $\tilde{\pi}$ is the apparent risk of disease in your study sample
- while $\pi$ is the actual risk of disease in the larger population

#### 15.1.1.5.3 Diminishing returns in unbalanced binary data

- This means that you don't get alot of 1's (the disease)
  - if you use unbiased sampling from the larger population.

So you can do "control to cases ratio"

If you have a sparse cases, you can sample it

#### 15.1.1.5.4 Multi-class Logistic Regression

- If you have multiple categorical responses then you use multi-class logistic regregression.

## Case-control sampling and logistic regression

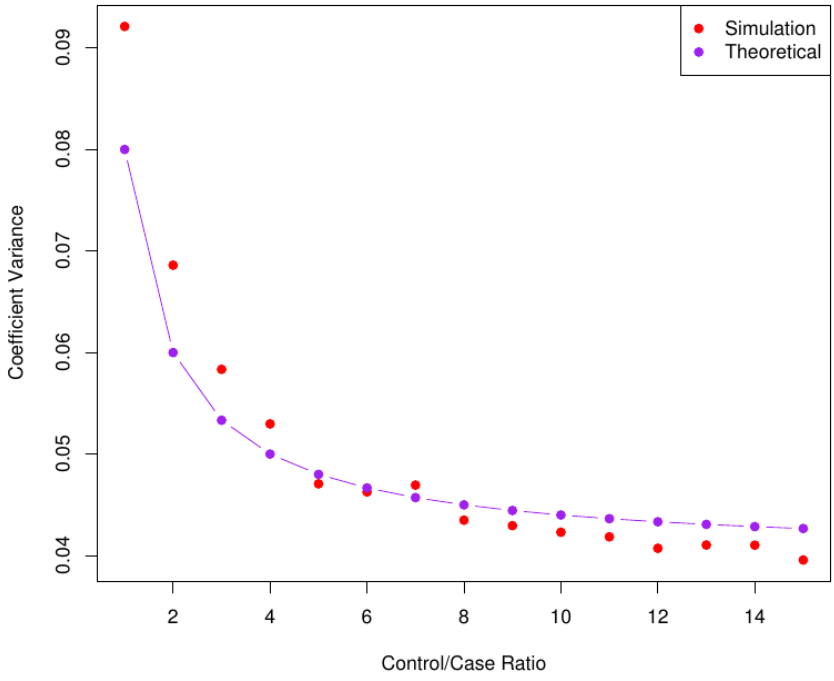
- In South African data, there are 160 cases, 302 controls — $\tilde{\pi} = 0.35$ are cases. Yet the prevalence of MI in this region is $\pi = 0.05$.
- With case-control samples, we can estimate the regression parameters $\beta_j$ accurately (if our model is correct); the constant term $\beta_0$ is incorrect.
- We can correct the estimated intercept by a simple transformation

$$\hat{\beta}_0^* = \hat{\beta}_0 + \log \frac{\pi}{1-\pi} - \log \frac{\tilde{\pi}}{1-\tilde{\pi}}$$

- Often cases are rare and we take them all; up to five times that number of controls is sufficient. See next frame

Figure 4: Case Control Studies

## Diminishing returns in unbalanced binary data




Sampling more controls than cases reduces the variance of the parameter estimates. But after a ratio of about 5 to 1 the variance reduction flattens out.

Figure 5: Case Control Ratio

And here you use glmnet package

## Logistic regression with more than two classes

So far we have discussed logistic regression with two classes. It is easily generalized to more than two classes. One version (used in the R package `glmnet`) has the symmetric form

$$\Pr(Y = k|X) = \frac{e^{\beta_{0k}+\beta_{1k}X_1+...+\beta_{pk}X_p}}{\sum_{\ell=1}^{K} e^{\beta_{0\ell}+\beta_{1\ell}X_1+...+\beta_{p\ell}X_p}}$$

Here there is a linear function for *each* class.
(The *mathier* students will recognize that some cancellation is possible, and only $K - 1$ linear functions are needed as in 2-class logistic regression.)

Multiclass logistic regression is also referred to as *multinomial regression*.

#### 15.1.1.5.5 Discriminant Analysis

- A different form of classification analysis.

Model the distribution of $X$ in each of your classes $Y$.

Then use Bayes theorem to flip around and get $Pr(Y|X)$

You can get $Pr(Y = k|X = x)$ by knowing $Pr(X = x|Y = k)$ and adding in "priors"

- $Pr(Y = k)$ is called the marginal probability or prior probability of $Y = k$
- And you have the marginal probability of $Pr(X = x)$

# Bayes theorem for classification

Thomas Bayes was a famous mathematician whose name represents a big subfield of statistical and probablistic modeling. Here we focus on a simple result, known as Bayes theorem:

$$\Pr(Y = k|X = x) = \frac{\Pr(X = x|Y = k) \cdot \Pr(Y = k)}{\Pr(X = x)}$$

One writes this slightly differently for discriminant analysis:

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}, \quad \text{where}$$

- $f_k(x) = \Pr(X = x|Y = k)$ is the *density* for $X$ in class $k$. Here we will use normal densities for these, separately in each class.
- $\pi_k = \Pr(Y = k)$ is the marginal or *prior* probability for class $k$.

Logistic regression works well when you don't have strong predictors,

- i.e. for very complex systems with lots of predictors and interactions

Discriminant analysis is bettter used

- for cases where the classes are well separated and predictors are strong.

# Why discriminant analysis?

- When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable. Linear discriminant analysis does not suffer from this problem.
- If $n$ is small and the distribution of the predictors $X$ is approximately normal in each of the classes, the linear discriminant model is again more stable than the logistic regression model.
- Linear discriminant analysis is popular when we have more than two response classes, because it also provides low-dimensional views of the data.

- $\mu_k$ is the mean in class k
- $\sigma_k^2$ is the variance in class k

#### 15.1.1.6 Notation Sidebar

- We have equations
  - For models The $Y$'s $X$'s,
  - The values of predictors $x_i$ and $y_i$'s,
  - The expected values of responses $\hat{y}$

We have the `lm` model notation.

We have

- $\mu$ is means,
- $\sigma^2$ are variances,
- $\pi$ are probabilities

---

#### 15.1.1.7 Machine Learning, Deep Learning and AI

- We'll start some readings on Deep Learning
  - Deep Learning is a type of Machine Learning
  - Done using large Neural Networks
  - And applied to very large datasets

#### 15.1.1.7.1 Google, through its Deep Mind unit, has been doing cool things

- Google AI Blog
- Google DeepMind Blog
- Alpha Go, Alpha Fold, Alpha Zero
  - Using Deep Learning and Reinforcement Learning

#### 15.1.1.7.2 Next semester we'll start learning GPU based ML/DL

- So Machine Learning has many methods
- Deep Learning is mostly done with Neural Networks
  - Relys on GPUs instead of CPUs
  - Mostly made by Nvidia

#### 15.1.1.7.3 TensorFlow

- Is Google's Open Source ML Library

Keras is a higher level ML interface

- Works with TensorFlow
  - And with Theano
  - And with CNTK
- And makes for builing simpler Deep Learning models
  - With these ML Libraries

#### 15.1.1.7.4 TensorFlow/Keras can be used from R

- But Rstudio has made TensorFlow from R Package
- And R Interface for TensorFlow
- And R Interface for Keras

#### 15.1.1.7.5 So ML/DL Benefit in speed from GPUs

- Nvidia is the biggest GPU for ML producer
  - Keras and TensorFlow only run on Nvidia GPUs
    - And this requires rather complex setup to compute on GPU
  - AMD makes GPUs for gaming
    - But these haven't been developed into ML GPUs yet
  - And Intel GPUs don't work for ML
  - Their "GPUs" are light weight, only to drive display screens
  - Mac doesn't sell computers with real GPUs

Also Operating Systems

- Windows does not ship with code development tools
  - Like compilers and development environments
- Apple hasn't been catering to Coders
  - So doesn't both making Nvidia GPU computers
  - They prefer the cheaper Intel and AMD GPUs
  - That don't work for ML
- Apple (like Microsoft) have been removing development tools from OSX
  - This is why you have to install XQuartz and HomeBrew on a Mac
  - To do data science

So Linux is the main Operating System used for Machine Learning

- Either Debian/Ubuntu/Kubuntu or Red Hat/CentOS/Fedora

- So setting up GPUs and TensorFlow/Keras
  - Is easiest in Linux

We will use CWRU's Markov: High Performance Computing (HPC) Cluster

- Runs on Red Hat RHEL7
- Has TensorFlow and Keras installed
- That Run on GPUs
- And has R and RStudio

So we'll learn some Linux (like using Git Bash)

- Using X2Go to graphically login to HPC
  - Get a compute node
  - Fire up Rstudio, and do your work
- You can also login to HPC
  - Using a command line interface (CLI)
  - by "ssh'ing" in
  - But you don't typically use graphical interface tools this way

You can also access GPUs and Tensorflow

- On Markov Data Science Cluster
- Or from a Kaggle account
- Or from Google CoLab