

Lecture 5b: Machine Learning in R

15.S60 Software Tools for Operations Research

Jerry Kung
MIT Operations Research Center
jkung@mit.edu

January 18, 2016

Outline

- 1 Linear and Logistic Regression
- 2 CART and Random Forests
- 3 Clustering
- 4 Support Vector Machines
- 5 Conclusion

Linear Regression

- We wish to find a linear function of the explanatory variables that best explains a single response variable
- To do this, we minimize the sum of the squared errors of the predictions of our function. The errors are known as *residuals*, and this method is also known as ordinary least-squares regression
- We can measure how well a linear model explains the data using its R^2 value
 - Beware of overfitting!
- Regularized methods are also popular and have nice properties
 - Ridge regression (additional penalty of L_2 norm of the coefficients)
 - Lasso (additional penalty of L_1 norm of the coefficients)

Logistic Regression

- Used for binary classification (e.g., predict whether or not a passenger survived)
- Uses a logistic function to predict the probability of a class:

$$\mathbb{P}(\hat{y} = 1) = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + \dots + b_k x_k)}}$$

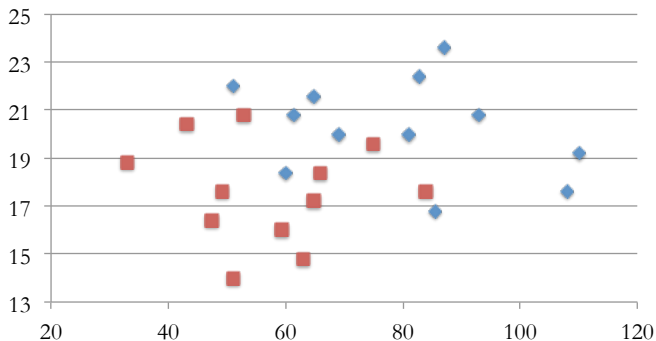
- Think about the term in the exponent as a linear regression; the logistic function then translates the estimate from the linear regression to a value between 0 and 1
- We then use a threshold value on the output of the logistic function
 - Often, this value is 0.5, meaning that we predict the class with the higher probability
 - Sometimes other threshold values may be more appropriate

Outline

- 1 Linear and Logistic Regression
- 2 CART and Random Forests**
- 3 Clustering
- 4 Support Vector Machines
- 5 Conclusion

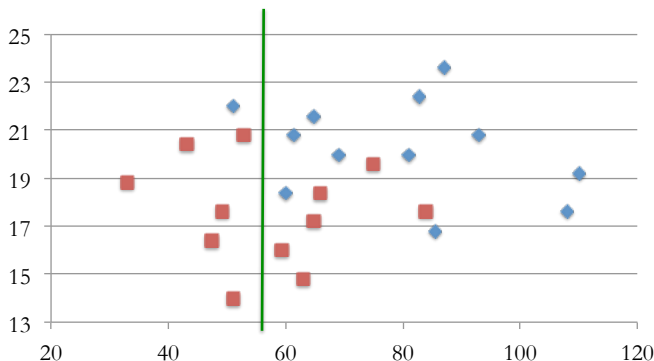
Classification and Regression Trees (CART)

- Make sequential splits on independent variables (was the passenger male? if yes, then consider age; if no, then consider class, and so on)
- Splits are made to make the “buckets” as “pure” as possible, in a greedy fashion



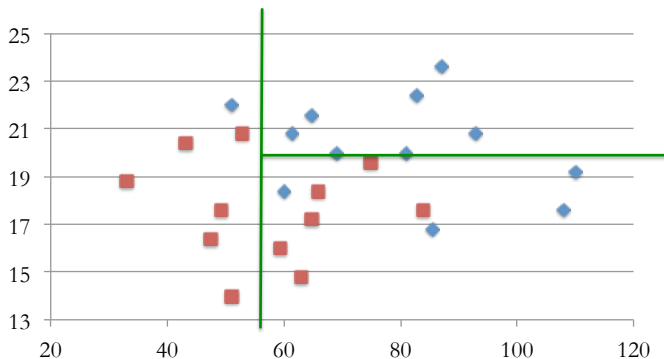
Classification and Regression Trees (CART)

- Make sequential splits on independent variables (was the passenger male? if yes, then consider age; if no, then consider class, and so on)
- Splits are made to make the “buckets” as “pure” as possible, in a greedy fashion



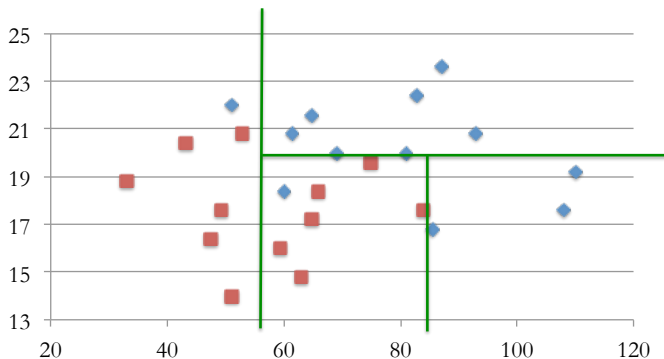
Classification and Regression Trees (CART)

- Make sequential splits on independent variables (was the passenger male? if yes, then consider age; if no, then consider class, and so on)
- Splits are made to make the “buckets” as “pure” as possible, in a greedy fashion



Classification and Regression Trees (CART)

- Make sequential splits on independent variables (was the passenger male? if yes, then consider age; if no, then consider class, and so on)
- Splits are made to make the “buckets” as “pure” as possible, in a greedy fashion



Classification and Regression Trees (CART)

- You can usually build a tree that fits the data exactly (except when two identical data points have differing classifications)
- Thus, we have to limit the power of the algorithm so that we don't overfit (strong performance on training data, but poor ability to generalize to test data and beyond)
 - `minbucket` requires each terminal leaf to have at least some number of points
 - `minsplit` will require a minimum number of points in a bucket before a split can be made
 - Can also limit the number of splits

Classification and Regression Trees (CART)

When might CART perform poorly?

Classification and Regression Trees (CART)

- Predictions made by CART trees can be multi-class categorical (e.g., yes or no; red, green, or blue, etc.). In this case, the prediction for a test data point is just the label of the majority of the points in that “bucket”
- Predictions can also be continuous (e.g., price, distance, etc.). In this case, the prediction for a test data point is just the average of the points in that “bucket”

Random Forest

- The random forest algorithm builds many CART trees that are uncorrelated
- For each CART tree, select only a subset of the data points on which to regress
- When making splits, only choose a subset of attributes that are permitted to be split upon (changes at each iteration of splits)
 - Defaults in R (which can be changed using `mtry`) are \sqrt{m} for classification and $\frac{m}{3}$ for regression
- Prediction is the majority “vote” for classification and the mean of all trees for continuous outcomes

Outline

- 1 Linear and Logistic Regression
- 2 CART and Random Forests
- 3 Clustering**
- 4 Support Vector Machines
- 5 Conclusion

Types of Clustering

- Hierarchical (Agglomerative) Clustering
 - Each data point starts out as its own cluster. Then merge based on a given criterion until there is only one cluster remaining.
 - No need to specify the number of clusters, since you can visualize the hierarchy.
- K-means Clustering
 - Begin by specifying the centroids of k clusters. Assign each data point to the nearest cluster. Then re-center the resultant clusters and iterate.
 - Less computationally intensive than hierarchical clustering, so better for larger data sets.
- Many other types of clustering as well

Distance Metrics

- In order for us to cluster data points, we need to define a notion of distance between data points.
 - Today, we will use Euclidean distance (L_2 norm)
 - Manhattan (L_1 norm) and Maximum coordinate (L_{inf} norm) are common as well
 - Often is highly sensitive to scale (we can normalize by subtracting the mean and dividing by the standard deviation)
- We also need to define distances between two clusters.
 - Today, we will use centroid distance
 - Maximum and minimum cluster distances can be used as well

Hierarchical Clustering vs. K-means

- Advantages of Hierarchical Clustering:
 - No need to specify number of clusters in advance (use dendrograms to visualize)
 - Clusters are nested (you can see the hierarchy)
- Advantages of K-means Clustering:
 - Faster (important for large data sets)
 - Less influenced by choice of distance metric

Outline

- 1 Linear and Logistic Regression
- 2 CART and Random Forests
- 3 Clustering
- 4 Support Vector Machines**
- 5 Conclusion

Support Vector Machines

- At a high level, SVMs determine a decision boundary that maximizes the “margin” (which, roughly, is the distance from the boundary to the support vectors)
- The objective is to maximize this margin, while the constraints are to obey the labels on the points
- A nonlinear optimization problem, but can be solved efficiently (if interested, see Wikipedia article on SVMs)
- Using kernels, decision boundaries can be extended to polynomials and beyond (Gaussian, or radial basis, kernels can be constructed so that any training set can be classified correctly)

Outline

- 1 Linear and Logistic Regression
- 2 CART and Random Forests
- 3 Clustering
- 4 Support Vector Machines
- 5 Conclusion**

Thanks for listening!

- Special thanks to Allison O'Hair
- Solutions can be found at Git <https://github.com/joehuchette/OR-software-tools-2016/tree/master/5-excel>
- Please fill out feedback forms!
- Any questions? Feel free to e-mail jkung@mit.edu