Chapter 8

Classification

Classification

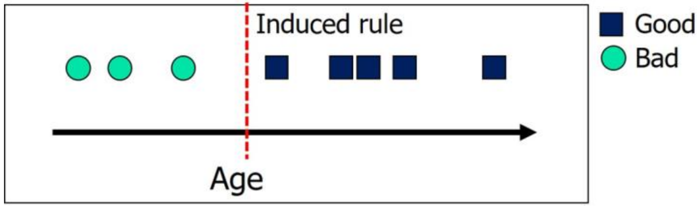
- One of the most frequent task in analytics
  - Without paying attention, we are all the time classifying things
  - We perform a classification task when:
    - Marking a comment as rude or polite
    - Adding someone to our social network
    - Telling our child if an animal in the zoo is a bear, bird, cat etc.
    - Reading numbers from a sheet of paper
- The main difference from Regression is that in classification the target is discrete

Classification

- Classification Task
  - Predictive task where a label to be assigned to a new, unlabeled, object, given the value of its predictive attributes, is a qualitative value representing a class or category.
  - Classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.

Example

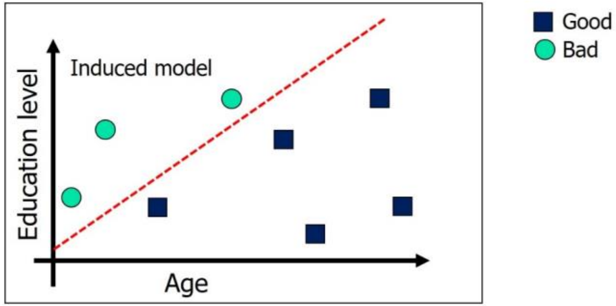
Name	Age	Company
Andrew	51	Good
Bernhard	43	Good
Dennis	82	Good
Eve	23	Bad
Fred	46	Good
Irene	29	Bad
James	42	Good
Lea	38	Good
Mary	31	Bad



If age < 32
Then company is Bad
Else company is Good

Example

Name	Age	Education level	Company
Andrew	51	1.0	Good
Bernhar d	43	2.0	Good
Dennis	82	3.0	Good
Eve	23	3.5	Bad
Fred	46	5.0	Good
Irene	29	4.5	Bad
James	42	4.0	Good
Lea	38	5.0	Bad
Mary	31	3.0	Good



If person > decision border
Then company is Bad
Else company is Good

Classification Algorithms

- Dozens of algorithms exist and a lot of them have many variations
- The algorithms can be *classified* into 4 categories
  - **Distance-based algorithms**
  - **Probability-based algorithms**
  - Search-based algorithms
  - Optimization-based algorithms

Classification Algorithms: Distance-based

- **Distance-based algorithms**
  - **K-nearest Neighbor**
  - Case-based Reasoning

Classification Algorithms: Probability-based

- **Probability-based algorithms**
  - **Logistic Regression**
  - Naïve Bayes

Classification Algorithms: Search-based

- **Search-based algorithms**
  - Decision Tree
  - Random Forest

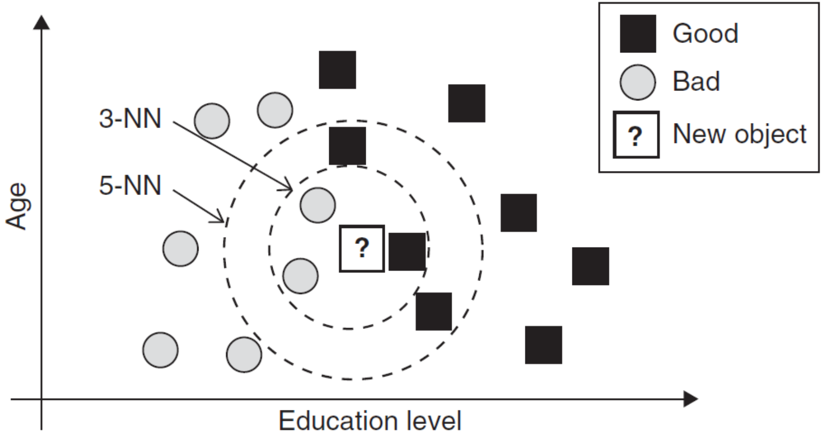
Classification Algorithms: Optimization-based

- **Optimization-based algorithms**
  - Support Vector Machines
  - Artificial Neural Networks

K-nearest Neighbor Algorithm

Algorithm K-NN test algorithm.

INPUT D_{train}, the training set
INPUT D_{test}, the test set
INPUT d, the distance measure
INPUT x_i objects in the test set
INPUT K, the number of neighbors
INPUT n, the number of objects in the test set
for all object x_i in D_{test} **do**
    for all object x_j in D_{test} **do**
        Find the k objects from D_{train} closest to x_i according to the chosen distance measure d
    Assign x_i the class label most frequent in the k closest objects



Note: The algorithm can be easily transformed into a Regressor if we simply return the mean of the k closest object's target attribute

K-nearest Neighbor Algorithm

- **Pros**
  - Its simplicity
  - Good predictive power in several problems
  - It is inherently incremental
- **Cons**
  - k-NN can take a long time to classify a new object
  - The use of only local information to classify new objects
  - Sensitive to the presence of irrelevant attributes and outliers
  - **Predictive quantitative attributes need to be normalized**

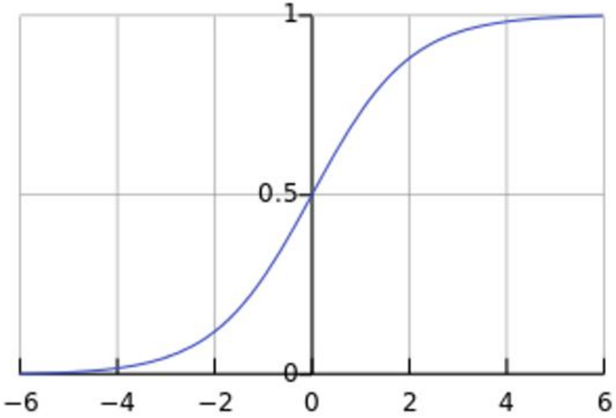
Logistic Regression Algorithm

- Many problems of classification are not deterministic
  - The relationship between input attributes and class is probabilistic
  - For example card games, sports bets, etc.
- Despite the misleading name this is a classifier not a regressor
- Built on top of linear regression
- „Classification via Regression”

Logistic Regression Algorithm

- The problem: $p(y=1 \mid \boldsymbol{x};\boldsymbol{\theta})$
- As we can see this is a binary problem
- Linear Regression: $\hat{y} = \mathbf{wx}$
- Logistic Regression: $\hat{y} = \sigma(\mathbf{wx})$
- Sigmoid/logistic function:

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



Logistic Regression Algorithm

- **The Objective Function for Linear Regression**:

$$E = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 + \lambda\sum_{i=1}^{p}(w^{(i)})^2$$

- **The Objective Function for Logistic Regression**
  - Cost for the i-th instance:

$$cost(\hat{y}^{(i)}, y^{(i)}) = \begin{cases} -log(\hat{y}^{(i)}) & \text{if } y^{(i)} = 1 \\ -log(1-\hat{y}^{(i)}) & \text{if } y^{(i)} = 0 \end{cases}$$

$$= -y^{(i)}log(\hat{y}^{(i)}) - (1-y^{(i)})log(1-\hat{y}^{(i)})$$

  - Total cost:

$$-\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)}log(\hat{y}^{(i)}) + (1-y^{(i)})log(1-\hat{y}^{(i)})\right)$$

  - This is called cross-entropy

Logistic Regression Algorithm

- **How to do Logistic Regression with multiple classes?**
  - The easies solution is to train an independent model for each of the class labels
  - This is called One-vs-Rest algorithm
  - The problem is this assumes the classes are distinct and non-related
- **However Logistic Regression is using a linear model we can extend the data with polynomial features as we did with Linear Regression previously**

Measuring predictive performance

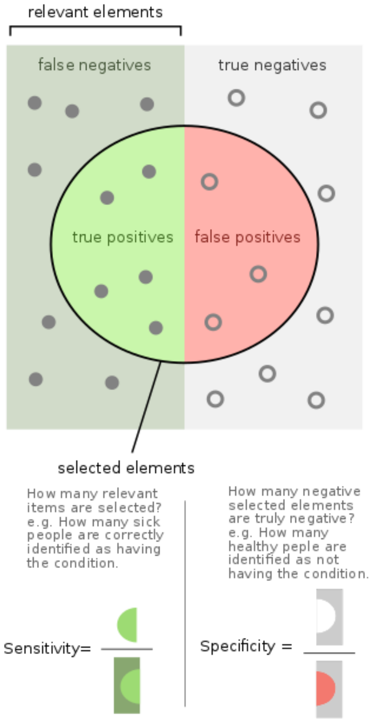
- Assess predictive performance of a classification model
  - How frequent the predicted labels are the true class labels?
  - Model predictive performance must be better than predicting in the majority class
    - Class with the largest number of objects

Measuring predictive performance

- Confusion matrix reports the predictive performance of a binary classifier
  - True class
    - Positive class
    - Negative class
  - Predicted class
  - Each cell contains the count
  - Can be easily extended to multiclass problems

Predicted class \ True class	p	n
P	True positives (TP)	False positives (FP)
N	False negatives (FN)	True negatives (TN)

Measuring predictive performance



$$\frac{FP}{FP+TN}$$ False positive rate (FPR) = 1-TNR

$$\frac{FN}{TP+FN}$$ False negative rate (FNR) = 1-TPR

$$\boxed{\frac{TP}{TP+FN}}$$ True positive rate (TPR), also known as recall or sensitivity

$$\boxed{\frac{TN}{TN+FP}}$$ True negative rate (TNR), also known as specificity

$$\boxed{\frac{TP}{TP+FP}}$$ Positive predictive value (PPV), also known as precision

$$\boxed{\frac{TN}{TN+FN}}$$ Negative predictive value (NPV)

$$\frac{TP+TN}{TP+TN+FP+FN}$$ Accuracy

$$\frac{2}{1/\,precision+1/\,recall}$$ F1-measure

Literature, further reading

- Multinomial Logistic Regression https://en.wikipedia.org/wiki/Multinomial_logistic_regression
- Further methods and comparison https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html
- http://biointelligence.hu/pdf/02-from-linear-regression-to-deep-learning.pdf (Special thanks to Krisztian Buza for allowing me to use materials from this presentation)
- https://www.deeplearningbook.org/contents/ml.html

Questions?