Machine Learning

Lesson 3

Instance Based Learning, Feature Reduction, Collaborative Filtering,

Lecturer: Dr. Msagha J Mbogholi, PhD

Flashback from Lesson 2

- In instance-based learning, a new instance which we wish to classify is compared to the instances in the training data.
- In model-based learning, a model is built from the training examples.
- An overfit model is one that has learnt the training data so well that it underperforms when applied to unseen data.
- In all linear regression modelling there is only one outcome (y), while there may be one or more input/predictor/independent variables (x).
- A decision tree is a tree like structure like a flowchart that starts with a desired outcome, and flows down considering each option (decision) and what it takes to fulfill that option, until all the options have been exhausted and an outcome determined.

Content

- Instance Based Learning
- Feature (dimensionality) Reduction
- Collaborative Filtering

Part 1

Instance Based Learning

1.1 Introduction

- In lesson 2 the concept of instance based learning and model based learning were introduced.
- Recall that it was stated that in instance based learning , a new instance which we wish to classify is compared to the instances in the training data.
- Further the new instance takes the class of the instances that are most similar to it.
- Model based learning is also referred to as parametric learning; this is because with model based learning there are training examples provided and these will be used to build the model. Any unseen data provided thereafter will use the existing model to test its performance. In a nutshell in model based learning there are a fixed number of parameters.
- Consider the example of simple linear regression described in lesson 2 as a good example of parametric model learning.
- This type of learning is also referred to as eager learning.

1.1 Introduction (cont'd)

- This is not the case with instance based learning.
- Recall from lesson 2 that with this kind of learning when a new instance is introduced it is compared to other instances in the training data and is classified accordingly.
- Thus in this case no model is developed and since processing is only done when a new instance is introduced, this type of learning is also described as lazy learning.
- The effect on memory is clear; it is quite memory intensive since for every new instance introduced the training data has to be recalled from memory and classification done according to the specification used.
- Further consider the cost involved in classifying each and every new instance; this is more so if the instances have several attributes to be considered.
- Due to their utilization of memory and instances as described, instance based learning is also referred to as non parametric or memory based.

1.2. Instance Based Density Estimation

- With every sample/instance (variable) in instance based learning a comparison of the sample to the training data has to be made.
- Since the algorithm will use an estimate in comparing new instances with the training data there is need to use a function to estimate that the new instance will be in the range required and thus be classified as belonging to that instance in the training data. A probability density function (PDF) helps to achieve this. In machine learning the PDF helps to calculate probabilities from random samples.
- The probability density function is defined as "a function of a continuous random variable, whose integral across an interval gives the probability that the value of the variable lies within the same interval" (Oxford.com)

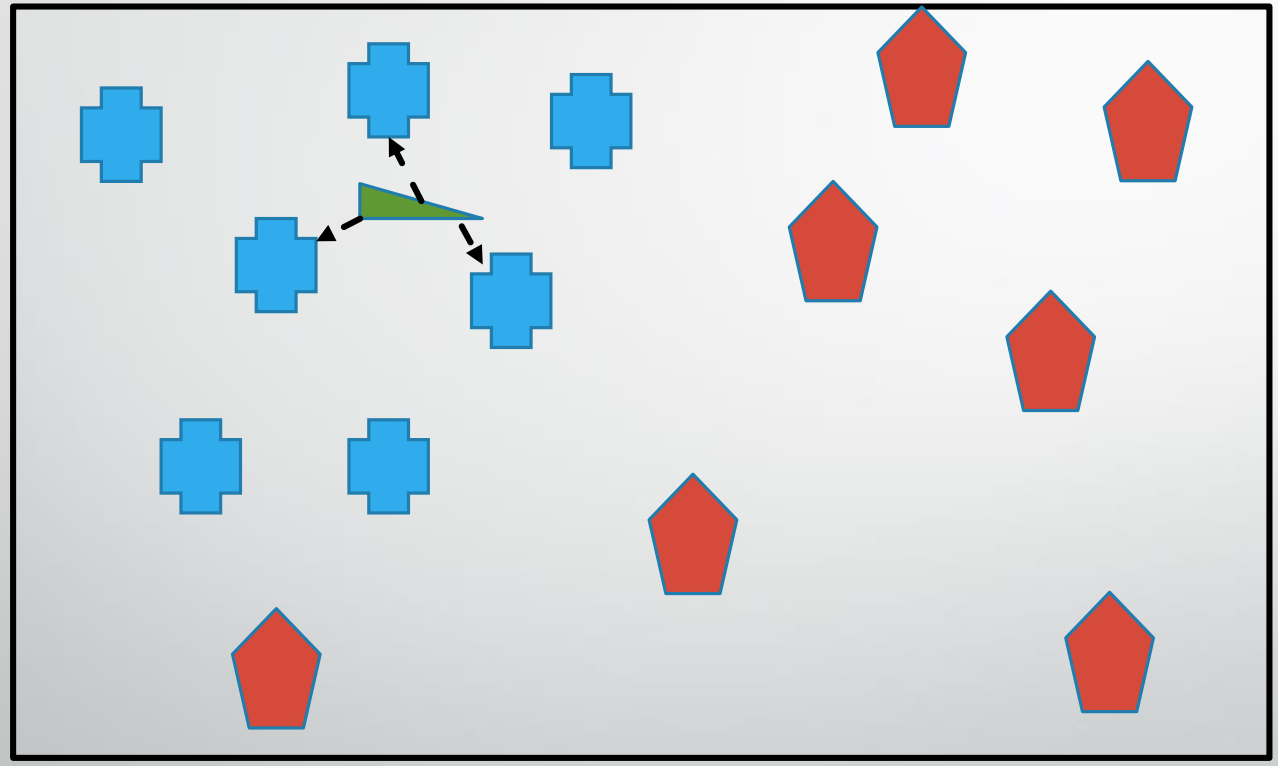
1.2. Instance Based Density Estimation (cont'd)

- The shape that the probability density takes is what is referred to as the probability distribution. The most common distribution is the normal distribution.
- The PDF helps to know whether a given instance lies within the range of likelihood or whether it can actually be classified as either an anomaly or an outlier.
- Instance based density estimation is what is used to approximate the PDF.
- Density estimation is done using histogram (histogram density estimation) and kernel density estimation (KDE).
- These two methods are described in detail in Alpaydin (2010).

1.3. K-nearest Neighbor (KNN)

- This is said to be one of the simplest algorithms to use in machine learning.
- It is a supervised learning algorithm.
- It is assumed that all instances belong to a plane of n – dimensions. For ease of explanation assume n to be 2 or 3 as these can be plotted easily on corresponding graphs.
- The nearest neighbor of any new instance is the shortest distance in a straight line (called the Euclidean distance) from it to a classified instance.
- The value of K is based on the number of neighbors that will be considered in classifying the new instance.
- Let us demonstrate this using 2 simple examples.

1.3 K nearest Neighbor (KNN) (cont'd)

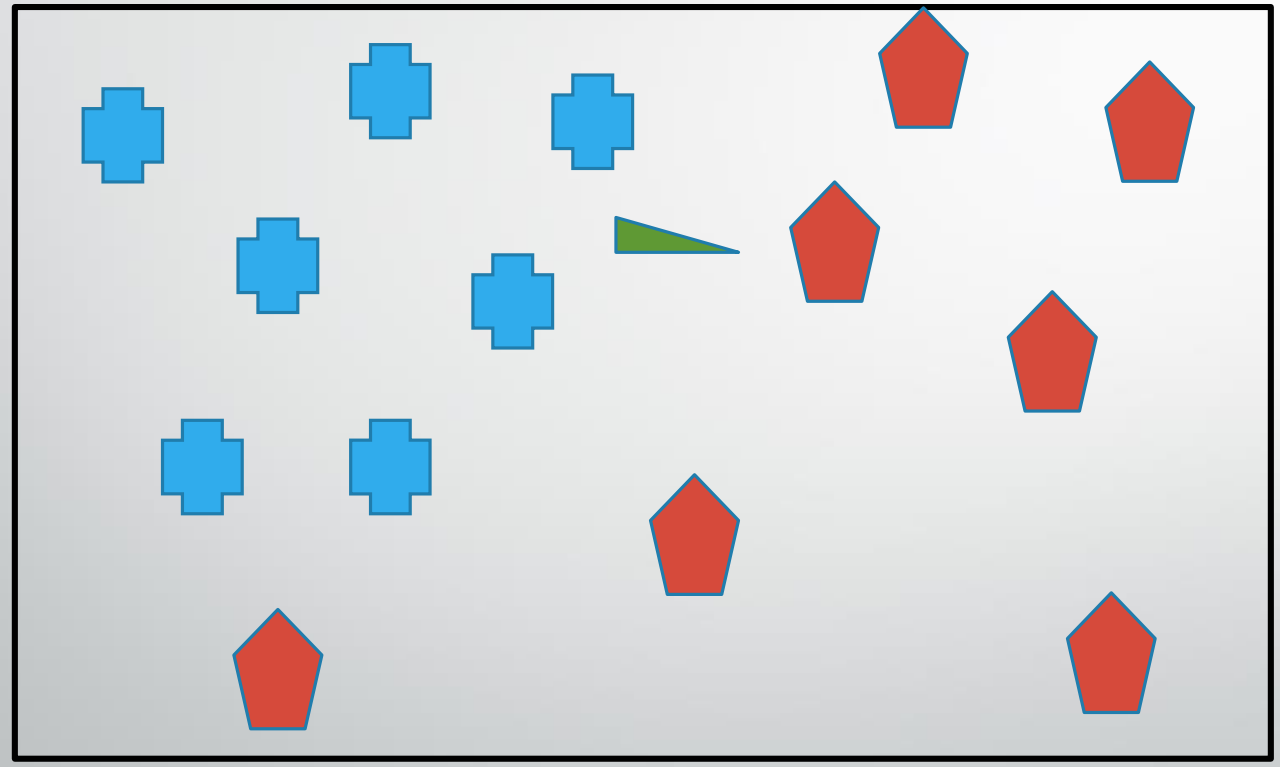


- Fig 1 KNN where K = 3

1.3 K nearest Neighbor (KNN) (cont'd)

- In Fig 1 there are three shapes presented. There are 7 red pentagons (RP) and 7 blue crosses (BC) already classified.
- We introduce a new instance in space that is green in color.
- In our learning we have specified that K = 3.
- In this case the learner will look at the green shape's 3 closest neighbors and find these are blue crosses (BC).
- The new instance is then classified as BC.

1.3 K nearest Neighbor (KNN) (cont'd)



- Fig 2. KNN where K = 2

1.3 K nearest Neighbor (KNN) (cont'd)

- Consider the scenario in Fig 2.
- Using the same logic and calculating the distance to the nearest neighbor what do you think the green figure will be classified as?
- It is thus very important to determine an optimum value of K to use in order to minimize errors in classification.
- This can be done by taking into consideration the error rate and validation errors with increasing values of K.
- Thus by separating the training and validation data an optimum value of K can be gotten from the validation error curve.

1.4 Curse of Dimensionality

- In our Fig 1 and Fig 2 examples we have examined data that has presumably just one attribute making it easier to classify using the KNN algorithm.
- Suppose that an instance has say n, attributes? Then this will mean that the distance from each attribute will have to be computed for the new instance.
- Supposing that an instance has 10 attributes. This will mean that KNN will be working in a 10 dimensional attribute space in classifying the new instance.
- Now suppose that out of these 10 attributes only 3 are truly relevant in determining the classification of the target function (the new instance)?

1.4 Curse of Dimensionality (cont'd):

- KNN will still apply itself to all the 10 attributes. Unfortunately it may be found that the irrelevant attributes are the ones that are nearest, leading to incorrect classification of the instance.
- This is referred to as the curse of dimensionality.
- How can this be ameliorated? Well one easy and practical way to do this is to give weights to each attribute.
- That way attributes that are more relevant will have more weights than those which are less relevant.
- When computation is done on the distances then the weighted distances are used which will result in more accurate classification.
- In an extreme case the attributes that are least relevant can be given a weight of zero, technically rendering them irrelevant in distance computation.

1.5 Locally Weighted Regression

- In lesson 2 linear regression was discussed and how it can be used to predict outcomes.
- What happens in a case where the data is non-linear, specifically in cases where non parametric learning is concerned? (the use of instances)
- Locally weighted regression is used in this instance.
- Visualized this means that the non linear data will be broken down into a series of lines with different gradients, right?
- When a new instance is introduced how is it determined which line it belongs to or use?
- The locally weighted regression algorithm assigns different weights to each of these "lines" such that if a given instance falls near a particular line it will have more weight than one which is far from it. Thus an outcome can be reasonably predicted.

1.6 Case Based Reasoning (CBR)

- CBR borrows from the key characteristics of instance based algorithms, namely:
  - Like instance based algorithms they only act based on new instances
  - Like KNN algorithms they classify new instances by examining how well they relate to existing instances.
- However, the methods they use to implement the above are more complex; for example they don't use points like in KNN, rather complex symbols and descriptions of the instances themselves
- They are used in a variety of fields such as engineering and manufacturing. Mitchell (1997) provides a few examples of applications of CBR.

Part 2

Feature (Dimensionality) Reduction

Introduction

- Feature (dimensionality) reduction is a topic of immense interest in machine learning.
- It is generally about reducing the dimensions of the instances so that we can draw more conclusive inferences.
- This topic is very wide and in the context of this course will be discussed in terms of feature selection, while reduction and transformation will be mentioned in context.
- At the end of the lesson the learner should use the provided references to dig deeper into the overall topic of dimensionality reduction.

Feature Selection

- Remember in section 1.4 of this lesson the curse of dimensionality was described?
- Feature selection is inspired by this curse; how?
- Feature selection purposes to remove the irrelevant and/or redundant features of the data.
- The key question is to determine what constitutes irrelevant/redundant data?
- The most important thing is that this data can be removed without affecting learning performance. There are two ways to go about this:

Feature Selection (cont'd)

- To check whether the removal of a feature say X, will affect learning performance; this will mean that there is another feature similar to this one , and therefore removing X will not affect learning performance.
- This is effectively achieved in one of two ways: the features can be weighted/ranked according to some criterion, OR a subset of features can be identified and used for the learning experience (as long as it does not affect learning performance)
- In both cases an algorithm is used to either rank the features based on some level (or threshold), or to create the subset that will contain an ideal number of features.
- Examples of such algorithms are Relief and wrapper algorithms respectively.
- Other examples of models used in feature selection are filters and embedded algorithms.
- Other important aspects of feature selection include models, search strategies, feature quality measures, and evaluation. (Liu & Motada, 1998)

2.1. Search Strategies

- Consider the issue of selecting the features to be used and the ones to be removed in a set of features; clearly a mechanism or strategy needs to be used in order to do this task.
- There are several strategies that are used in machine learning for searching; the most common ones are mentioned briefly:
- Wrapper methods - these include forward selection, backward selection, exhaustive selection, and recursive selection.
- Filter methods – these include information gain, chi square test, fisher's score and missing value.
- Embedded methods – these include random forest and regularization.

2.2 Evaluation

- The other issue is the evaluation of feature selection; how do we evaluate whether the selection method used was the optimum one?
- Clearly there is no one size fits all as far as choice of algorithm is concerned; it all depends on the scenario itself.
- One way of evaluating is to determine the effect before and after selection, i.e. did the algorithm achieve its intended objectives? Liu and Motoda (2008) suggest that number of selected features, time, scalability, and learning model's performance are also suitable parameters to use in evaluating the effectiveness of the feature selection algorithm used.

2.2 Evaluation (cont'd)

- Another way of evaluating is to compare two or more algorithms for the given scenario and see how well the learner performs using either of them.
- Care must also be taken to prevent overfitting and/or underfitting; different theorems abound on how to achieve this.
- One theory is to separate the learning data from the data used in feature selection; this is a paradox in itself since in most situations the same dataset used in selection is the same one used in learning. The reality is that the learner needs more training data for the learning in order to achieve optimal performance.
- Jakulin and Bratko (2004) and Zhao and Liu (2007) examined the issue of feature interaction. Feature interaction occurs in those scenarios where two features together can effectively classify a variable but individually can not. The latter suggested the use of special data structures that can handle the feature interaction effectively while Liu and Motoda (2008) suggest the use of randomized algorithms.

2.3 Unsupervised Feature Selection

- The challenge in unsupervised learning is that there is no labeled data; in supervised learning the search is guided by the labelled data in determining the features to adopt. What do you use in unsupervised learning?
- The dilemma is that in the absence of labeled data how do we select which features to use and which ones to drop/remove?
- A good example of unsupervised learning is clustering. This has been described in lesson and earlier in this lesson when describing KNN.
- Recall the curse of dimensionality? It is very applicable in this scenario since we are not sure which features to use and which ones to leave out.
- Thus it is preferred to use all the data; however, the reality is that not all the data is relevant due to issues such as redundancy and no value added. Consequently by using high volumes of data in the n-dimension space the curse of dimensionality comes into play.
- The goal of feature selection as defined by Dy and Brodley (2004, as cited by Liu and Motada (2008)) is "to find the smallest feature subset that best uncovers "interesting natural" groupings (clusters) from data according to the chosen criterion."

2.3 Unsupervised Feature Selection (cont'd)

- The best way to reduce the effect of the curse of dimensionality in unsupervised learning like clustering, is to reduce the n-dimensionality space. There are two documented ways to do this:
- Feature selection – using this technique the dimensionality is reduced by using subsets of the original space as opposed to using the whole space, thus retaining their original meaning.
- Feature transformation – using this technique a function is applied which reduces the dimensions; however, the original meaning of the features is not kept, and there is need to go further and extract the new (reduced) dimensions.
- When performing feature selection on the data in a cluster one may choose to use a single subset of data features encompassing all the data (global) or different sets of data features per cluster.
- The two factors that are of most importance is what the designer deems as relevant and redundant, since these two will mean different things to different designers.
- In terms of the methods used these are the same as those used in supervised feature selection; these are described in the next section.

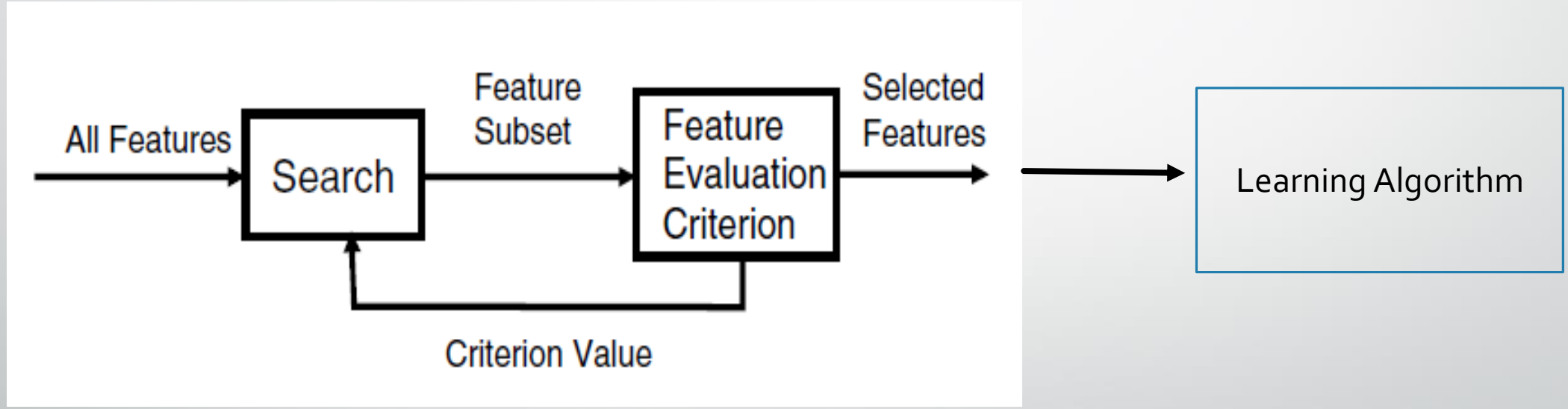
2.4 Feature Selection Methods

- Feature selection methods can be classified as belonging to one of three groups:
  - Filter methods
  - Wrapper methods
  - Embedded methods

2.4.1 Filter Methods

- With this method the data is filtered before it reaches the learning algorithm.
- They work on the input data directly without considering what learning algorithm will be used, and thus they are more like a pre-processing mechanism for the input data.
- The filter uses different criteria to remove what it considers redundant or irrelevant.
- On one hand they have the advantage of requiring less computational time due to the design of the model; on the other hand since they do not take into account the model being used it is considered a disadvantage to use.
- A very popular algorithm that uses this method is Relief.
- Fig 3 shows how the working of the filter method.

2.4.1 Filter Methods



- Fig 3. Filter method. Adapted from Liu and Motada (2008)

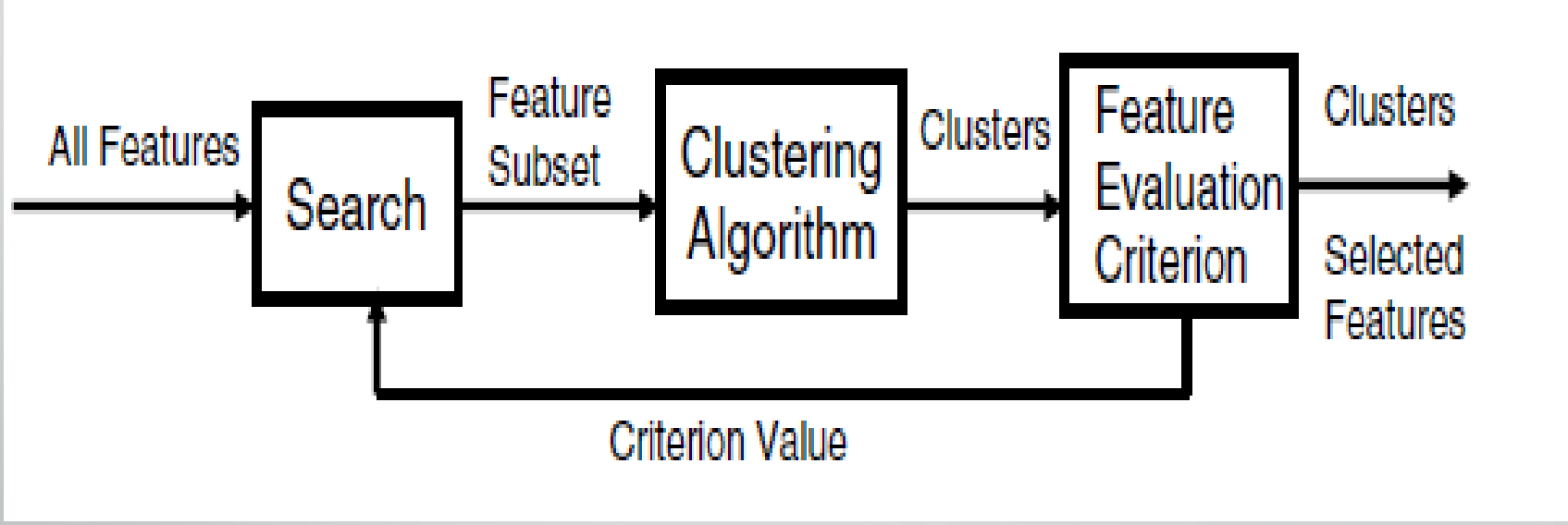
2.4.1 Filter Methods

- Techniques used by filter methods include:
- Information gain – this technique is used in many areas such as CART (classification and regression tree) algorithm used in decision trees. It is the amount of information gained by a variable based on observation of another variable. In a supervised learning environment it can be used to measure how each variable is with respect to the target function. In decision trees it is used to measure how much information is gained with every split of the decision tree.
- Chi-square test – this test is used simply to observe the difference between expected results and observed results. In machine learning it is used to observe the difference between attributes and the target values; consequently those with the best difference (chi square) are selected.
- Fisher's score – used in supervised learning to measure how much information a variable gives with respect to a parameter of interest. The variables with the largest score are selected.
- Missing Value ratio – it is a measure of the missing values in each column vs the total number of observations. Depending on the cut off value chosen those with higher ratio are dropped.

2.4.2 Wrapper Methods

- These methods are different from the filter methods in that they involve the learning algorithm in the selection process; they are therefore algorithm dependent.
- They work by taking the subset of features and training the algorithm with each step as the wrapper technique is used.
- Since they are classifier (algorithm) dependent they may work well with one classifier but not so with another one.
- They are also computationally costly due to the many steps involved and number of attributes involved.
- They can be used for feature selection and reduction, as well as for other purposes like making adjustments to variables for an algorithm. See Staˊnczyk (2013).
- Fig 4 shows the working of a wrapper for a clustering (unsupervised) algorithm. Note the positioning of the clustering algorithm compared to the one in Fig 3.

2.4.2 Wrapper Methods



- Fig 4. Wrapper method (Liu & Motada, 2008)

2.4.2 Wrapper Methods

- Techniques used by wrapper methods include:
- Forward selection – this technique begins with an empty set and goes on adding attributes till when adding an attribute does not improve the performance of the learning function. Sequential Forward Selection (SFS) is a type of forward selection.
- Backward selection – it is the opposite of forward selection. It begins with an full set and removes attributes one at a time until removing an attribute does not improve the performance of the learning function. Once a feature has been removed, however, it can not be included again.
- Exhaustive selection – this technique uses the brute force approach; it tries every possible combination of attributes till the best combination is found.
- Other techniques include floating, branch and bound, and recursive selection.

2.4.3 Embedded Methods

- This method includes the selection of features in the learning algorithm itself. They also show considerable progress in terms of learning performance.
- Techniques include:
- Regularization – this technique purposes to reduce overfitting and underfitting by calibrating using a factor such that it can be applied to the different attributes. Thus when an attribute returns a value below the threshold it is removed.
- Random forest – it is used in both decision trees and regression to select features. More on this technique will be discussed in later lessons.

Part 3

Collaborative Filtering

3.1 Introduction

- A recommender system is a system that helps a user discover products or content online.
- Most of us come in contact with recommender systems on a regular basis.
- Examples of recommender systems include:
  - YouTube recommending videos
  - YouTube personalized ads
  - Instagram feed and friend suggestions
  - Snapchat friend suggestions
  - News sites showing stories that might interest you
  - And so on….

3.1 Introduction (cont'd)

- What is the value added of these recommender systems?
- Consider the following:
- The modern customer is faced with an overwhelming abundance of choice.
- We need to be guided through the maze of stuff that's out there to the things or content need. Online businesses aim to retain customers. If customers' needs are met, they will come back again and again.
- Online activity by all of us has provided online companies with a treasure trove from which to learn about us.
- Organisations like Facebook and LinkedIn highly value their recommender algorithms. They are company secrets
- However, the general approaches are the same.

3.2 Collaborative Filtering

- On Amazon, you might see a list of books under the heading, "Customers who viewed this item also bought…"
- In collaborative filtering recommendations are made to a person based on other people who are similar to them. A person might be considered similar to you if you liked reading some of the same books
- Then if there are books that person has read and liked, but you have not, they can be recommended to you.
- This is how collaborative filtering systems work.

3.3 Example

- On Amazon, users rate books on a 5 star system.
- Suppose Amazon has the following user data as depicted in table 1, on the ratings of two books.

	Snow Crash	**Girl with the Dragon Tattoo**
Amy	5 stars	5 stars
Bill	2 stars	5 stars
Jim	1 star	4 stars

Table 1. User data on books and ratings (Mbogho, 2019)

3.3 Example (cont'd)

- The same data can also be presented graphically as shown in Fig 5.
- This presents a different view of the data and enables us to how the two books relate to the users in a 2 dimensional space.

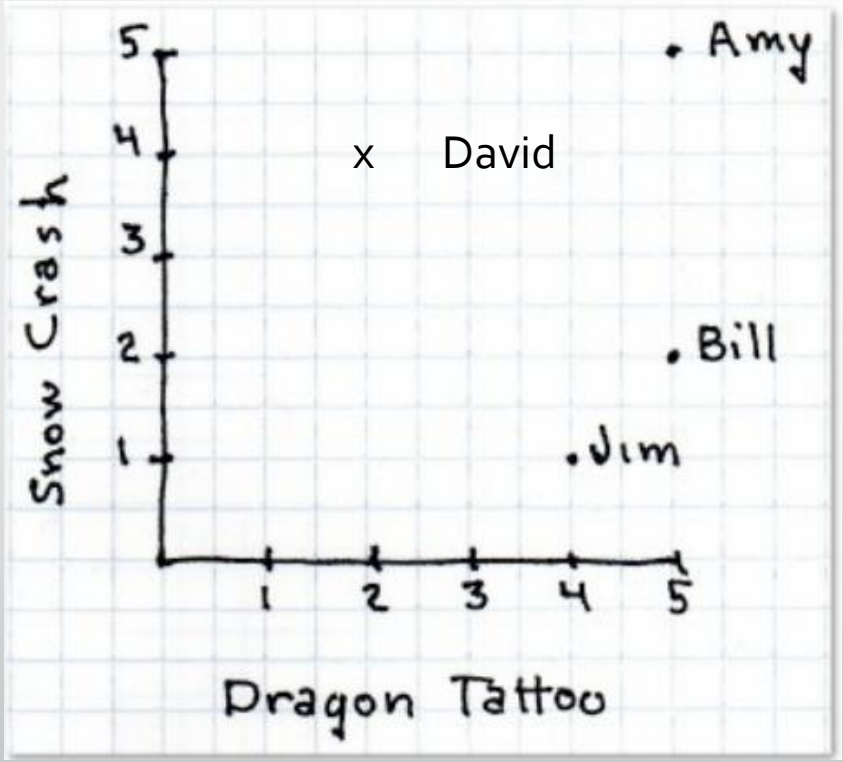


Fig 5. Data in 2 dimensional view. Adapted from Mbogho (2019)

3.3 Example (cont'd)

- An easy way to determine similarity between two users is by Manhattan Distance
- This distance counts how many steps separate the two individuals in the x-direction and in the y -direction and adds them together
- In 2D space,
- $MD((x1,y1), (x2,y2)) = |x1-x2| + |y1-y2|$

3.3 Example (cont'd)

- David is at location (2,4) marked X on the graph.
- His distance from Amy is (5-2) + (5- 4). = 4
- Suppose we wish to recommend a book to a new user, David who rated Snow Crash 4 and Dragon 2.
- We calculate the Manhattan Distance between David and all the other users and find that Amy is the closest.
- Suppose additional data shows that Amy read Introduction to Machine Learning, which David has not read, and rated it 4 stars?
- The book can now be recommended to David.

3.4 Class Prediction

- From existing data, we have learned David's interests and his similarity to another user.
- Thus we predict that David is in the same class as that user
- And we predict that David will like everything that is liked by members of this class.
- And we can recommend to him things liked by members of this class that he doesn't know about.
- Other ways of finding the class David will belong to include use of Euclidean distance and KNN. The methods are different but the principle is the same.

Summary

- Density estimation is done using histogram (histogram density estimation) and kernel density estimation (KDE).
- KNN is a supervised learning algorithm. The nearest neighbor of any new instance is the shortest distance in a straight line (called the Euclidean distance) from it to a classified instance.
- The curse of dimensionality may lead to incorrect classification due to irrelevant attributes being near to the instance due to many dimensions. As attributes increase so does the likelihood of incorrect classification.
- The search methods used for feature reduction include wrapper, filter and embedded methods.
- In collaborative filtering recommendations are made to a person based on other people who are similar to them.

References

- Alpaydin, E. (2010). *Introduction to machine learning*. MIT Press.
- *Feature selection techniques in Machine Learning - Javatpoint*. www.javatpoint.com. (n.d.). Retrieved April 15, 2022, from https://www.javatpoint.com/feature-selection-techniques-in-machine-learning
- Géron Aurélien. (2019). *Hands-on machine learning with scikit-learn and tensorflow concepts, tools, and techniques to build Intelligent Systems*. O'Reilly.
- Gurney, K. (2014). *An introduction to neural networks*. CRC Press.
- Jakulin, A., & Bratko, I. (2004). Testing the significance of attribute interactions. In *ICML '04: Twenty-First International Conference on Machine Learning*. ACM Press.
- Liu, H., & Motoda, H. (1998) *Feature Selection for Knowledge Discovery & Data Mining*. Boston: Kluwer Academic Publishers.

References

- Liu, H., & Motoda, H. (2008). *Computational methods of feature selection*. Taylor & Francis.
- Mbogho, A. W. (2019). *Introduction 3*. *Machine Learning*.
- Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.
- Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.
- Stańczyk Urszula, & Jain, L. C. (2015). *Feature selection for data and Pattern Recognition*. Springer.
- Verma, A. (n.d.). *Locally weighted regression*. Coding Ninjas CodeStudio. Retrieved April 15, 2022, from https://www.codingninjas.com/codestudio/library/locally-weighted-regression
- Zhao, Z. & Liu, H. (2007) Searching for interacting features. In *Proceedings of IJCAI - International Joint Conference on AI*, January.