Machine Learning Course Summary Continued

Notebook:	4th Year Project		
Created:	2018-09-29 8:39 PM	**Updated:**	2018-11-07 12:41 PM
Author:	moflanker@gmail.com		
Tags:	Machine Learning		
URL:	https://www.udemy.com/machinelearning/learn/v4/t/lecture/5765916?start=0		

About

This document continues on the last summary provided (https://www.evernote.com/l/AUJEW8ayNK5NJ4iG1vh1V3eueRefcxlxfxg/)

Estimation Time

This section was made for time management

Estimation time for the videos to learn the theory and implementations on udemy
** Time for external resources and coding practice to complement understanding is not included

- Regression
  - ☑ Simple Linear Regression
  - ☑ Multiple Linear Regression
  - ☑ Polynomial Regression
  - ☑ Support Vector Regression
  - ☑ Decision Tree Regression
  - ☑ Random Forest Regression
  - ☑ Random Forest Classification
  - ☑ Evaluating Regression Models Performance
- Classification
  - ☑ Logistic Regression
  - ☑ K-Nearest Neighbors (K-NN)
  - ☑ Support Vector Machine (SVM)
  - ☑ Kernel SVM

    - [x] Naive Bayes (Based on Bayes Theorem)
    - [x] Decision Tree Classification
    - [x] Random Forest Classification
    - [x] Evaluating Classification Models Performances
- Clustering
    - [x] K-Means Clustering (1 hour)
    - [x] Hierarchical Clustering (1 hour)
- Association Rule Learning
    - [x] Apriori (1.1 hour)
    - [x] Eclat (20 minutes)
- **Deep Learning**
    - [ ] Artificial Neural Networks ANN (160 minutes ~ 3 hours)
    - [ ] Conventional Neural Networks CNN (185 minutes ~ 4 hours)
    - [ ] Recurrent Neural Networks RNN

- ~~Dimensionality Reduction~~
    - [ ] ~~Principal Component Analysis PCA (75 min ~ 1.5 hours)~~
    - [ ] ~~Linear Discriminant Analysis LDA (30 min)~~
    - [ ] ~~Kernel PCA (20 min)~~
- ~~Reinforcement Learning~~
    - [ ] ~~Upper Confidence Bound (1.5 hours)~~
    - [ ] ~~Thompson Sampling (1 hour)~~
    - [ ] ~~used for gaming~~
- [ ] ~~Natural Language Processing (95 minutes - SKIP)~~
- [x] XGBoost (20 minutes)

Classification

SourceURL: https://www.udemy.com/machinelearning/learn/v4/t/lecture/5765916?start=0

Classification

Unlike regression where you predict a continuous number, you use classification to predict a category. There is a wide variety of classification applications from medicine to marketing. Classification models include linear models like Logistic Regression, SVM, and nonlinear ones like K-NN, Kernel SVM and Random Forests.

Understand and learn how to implement the following Machine Learning Classification models:

1. Logistic Regression

    - predict probability of true or false and classify accordingly
    - Linear problem
2. K-Nearest Neighbors (K-NN)
    - Classify according to the nearest subjects
    - vary k as an odd number
    - if K is even or sometimes odd, use the weighing technique
    - Weighing as how far subjects from the subject to be classified
3. Support Vector Machine (SVM)
4. Kernel SVM
5. Naive Bayes
    - same as logistic regression but for non-linear model
6. Decision Tree Classification
7. Random Forest Classification
    - A combination of decision trees for high performance (not always)
8. Evaluating Classification Models Performance
    - Cumulative Accuracy Profile (CAP) Curve
        - Accuracy Ratio
        - 90-100% accuracy, check over-fitting

Classification

Classification Model	Pros	Cons
Logistic Regression	Probabilistic approach, gives informations about statistical significance of features	The Logistic Regression Assumptions
K-NN	Simple to understand, fast and efficient	Need to choose the number of neighbours k
SVM	Performant, not biased by outliers, not sensitive to overfitting	Not appropriate for non linear problems, not the best choice for large number of features
Kernel SVM	High performance on nonlinear problems, not biased by outliers, not sensitive to overfitting	Not the best choice for large number of features, more complex
Naive Bayes	Efficient, not biased by outliers, works on nonlinear problems, probabilistic approach	Based on the assumption that features have same statistical relevance
Decision Tree Classification	Interpretability, no need for feature scaling, works on both linear / nonlinear problems	Poor results on too small datasets, overfitting can easily occur
Random Forest Classification	Powerful and accurate, good performance on many problems, including non linear	No interpretability, overfitting can easily occur, need to choose the number of trees

Machine Learning A-Z © SuperDataScience

Clustering

SourceURL: https://www.udemy.com/machinelearning/learn/v4/t/lecture/5765940?start=0

Clustering is similar to classification, but the basis is different. In Clustering you don't know what you are looking for, and you are trying to identify some segments or clusters in your data. When you use clustering algorithms on your dataset, unexpected things can suddenly pop up like structures, clusters and groupings you would have never thought of otherwise.

K-Means Clustering

- Choose centroid points, cluster/group based on least distance
- We are using Euclidean distance!! could be adjustable

Hierarchical Clustering

- **group similar objects building a Hierarchical graph**
- **select the number of lines to cross forming cluster of lower Hierarchical**

Clustering

Clustering Model	Pros	Cons
K-Means	Simple to understand, easily adaptable, works well on small or large datasets, fast, efficient and performant	Need to choose the number of clusters
Hierarchical Clustering	The optimal number of clusters can be obtained by the model itself, practical visualisation with the dendrogram	Not appropriate for large datasets

Machine Learning A-Z © SuperDataScience

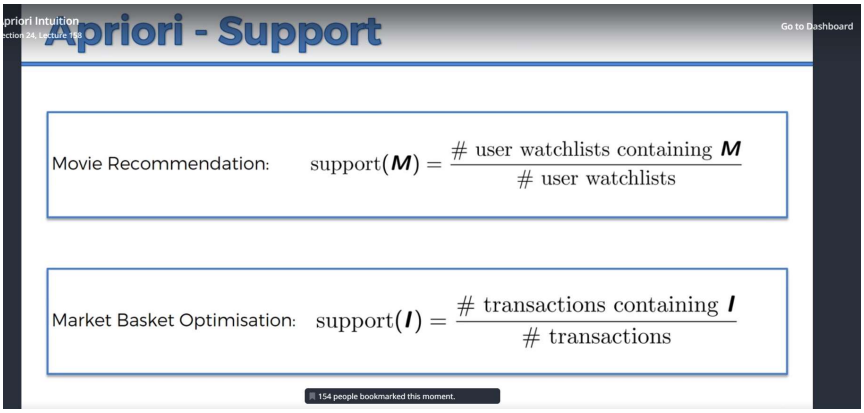
Association Rule Learning

Association Rule Learning

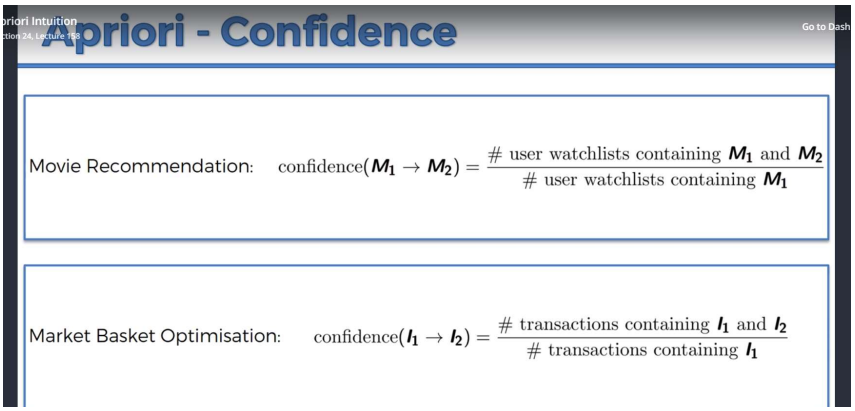
- method to discover interesting relationships among variables in large databases

Apriori Algorithm

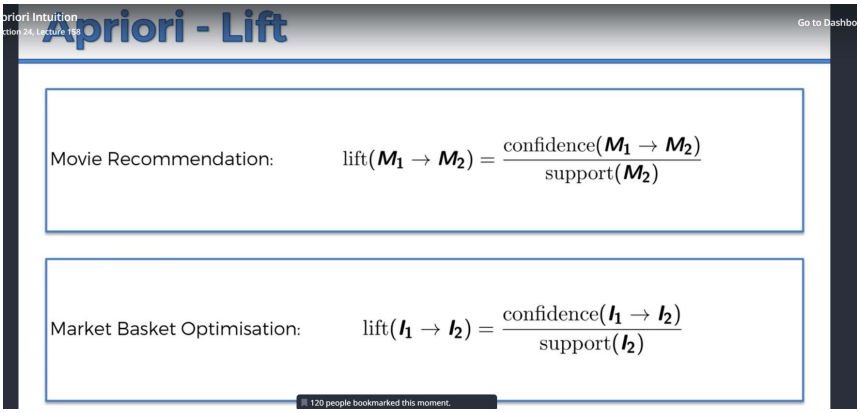
- support



- confidence



- lift



- Python library takes in a list of list
- Apriori algorithm arguments need some calculations and time to decide
    - it depends on the dataset you have

- R is more sophisticated for Apriori algorithm (statistics analysis)

https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html

Elcat Intiution

- dataset and correlation with an object ???

Reinforcement Learning

SourceURL: https://www.udemy.com/machinelearning/learn/v4/t/lecture/5985220?start=0

Reinforcement Learning is a branch of Machine Learning, also called Online Learning. **It is used to solve interacting problems where the data observed up to time t is considered to decide which action to take at time t + 1.** It is also used for Artificial Intelligence when training machines to perform tasks such as walking. Desired outcomes provide the AI with reward, undesired with punishment. Machines learn through trial and error.

Usually used for training games or robots.

Neural Network

SourceURL: https://www.youtube.com/watch?v=VrMHA3yX_QI&list=PLUl4u3cNGP63gFHB6xb-kVBiQHYe_4hSi&index=13

Neural Network

Learning methods is watching MIT videos on Neural Network to learn the essence of it and how to manipulate around it.

- Weights of nodes
- bias
- sigmoid or ReLu (for activation)
- learn underlying features in data

Processes steps

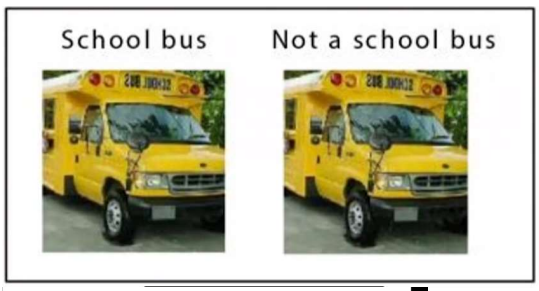
- Convolution
- Pooling
- Kernel layers
- Neural Net vectors
- Final output how likely it is an object (**cost function** - difference squared between expected and actual)
- Back propagation (**training the model by minimizing the cost function**)
- encoding of generalization (auto coding)

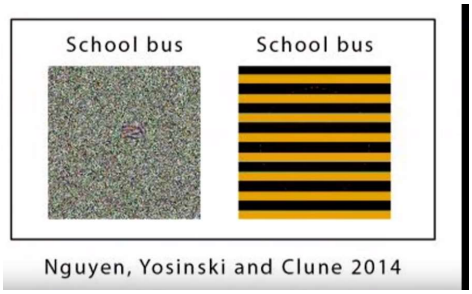
Z = f(X, W, T)
Z is a vector output
X is a vector input
We can change W weights and T threshold to get the desired output

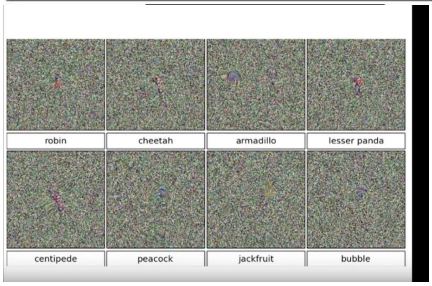
This is a extremely simple idea, all great ideas are simple. But why wouldn't we have more of them. Frequently, that simplicity involve finding couple tricks and couple of observation so usually human hardly go beyond one trick or one observation. But if you cascade few together sometime something miraculous falls out that looks in retrospect extremely simple. **Prof. Patrick Winston**

About Neural Network, there is nothing particularly special about them. It just happens to be a particular assembly of components that tends to reappear when anyone does this sort of NN stuff. Prof. Patrick Winston

Examples of NN false identify picture by manipulating pixels in images. Published papers on NN false identification







Bottom Line is these things are an engineering marvel and do great things but they don't see like we see. Prof. Patrick Winston

Algorithms that deals with time-series prediction and correlation

- **RNN** Recurrent Neural Networks - for time series specifically
  - learn underlying features in data (see Saliency maps) - for correlation study
- Hidden Markov Model ??
- **LSTM** long-short term memory

- **Saliency maps** - visualize RNN neural layers to perhaps find the **strongest correlation** to glucose level
  - this to be done if there is no strong correlation from the feature before modeling

    - for example we can independently investigate variable correlations to glucose level from the dataset we already have without the prediction step.

Note: Convolutional Neural Network (CNN) was mentioned in our meetings to be used in our project. However, I learned that for time series analysis, Recurrent Neural Network (RNN) is used which applies to our analysis. CNN is usually used for image recognition.
I'm done the ML course except Neural Networks (skipped the very unnecessary topics). Currently, I'm fully learning Neural Networks.

Note that Neural Network is a fairly large topic to learn and it is very interesting.

I was expecting to have an **expected data input format** this week to start developing a **prototype model**. For example, if I can be provided with what stress, calories in, calories out, sleep, and glucose level are going to be in time series graphs. This way I can start planning how I am going to tackle or break down the problem into smaller pieces. However, with the problems we are running into with data, we can discuss this next time we meet.

ML relies heavily in data preprocessing, and engineering features for models. **The algorithm doesn't change but we control what we input and how.** This is where engineering kicks in.
Good Machine Learning

- Cleaning data
- model selection
- engineering features
- design/select feature

Classify people

Big credits to the ML geniuses in my lab!!

Also, I want to discuss with Dr. Zied and Dr. Ibnkahla using **Linux** as it is super easier to use for **configuration and libraries** when working with ML. If so, I will ask them if they can provide us with a powerful PC to use that already has Linux.
With the huge number of data we are having, our personal PC will take a long time to run a test or a model.