IBM Machine Learning with Python - Coursera

Week 1 - What is Machine Learning

- Machine learning is the subfield of computer science that gives "computers the ability to learn without being explicitly programmed"

- Python Libraries (useful):
  - NumPy: Math library for effective computations
  - SciPy: Domain specific computation
  - Matplotlib: For plotting
  - pandas: High performance, easy to use data structures
  - SciKit learn: For ML ← Course focus!

- Classification: Discrete class labels or categories (Supervised)
  Regression: Continuous values

- Supervised learning: Predicting based on labeled data
  Unsupervised learning: Predicting based on unlabeled data
  Model works on it's own to discover info
  → Clustering
  → Fewer evaluation methods
  → Less controlled environment

Week 2 - Intro to Regression

- Regression: Predicting a continuous value

- X: Independent Variables (explanatory) - causes of these states
  Y: Dependent Variable - value we are trying to predict

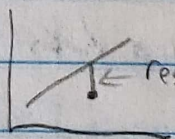
- X: Could be categorical or continuous
  Y: Has to be continuous

- Simple Regression: Predict CO_2 emission vs Engine Size
  Multiple Regression: Predict CO_2 emission vs Engine Size & Cylinders

- Simple R: $\hat{y} = \theta_0 + \theta_1 x_1$
  Linear
  $\hat{y}$: response variable
  x_1: single predictor
  θ_0: Intercept
  θ_1: Slope

- Mean Squared Error (MSE) $= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

  Residual error of each individuals squared then summed up and divided by total.

- Can calculate $\theta_1 = \frac{\sum_{i=1}^{s}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{s}(x_i - \bar{x})^2}$ & θ_0 by substituting

- Pros of linear regression: • Very fast
  • No parameter tuning
  • Easy to understand & highly interpretable

Accuracy

- Training Accuracy: Percentage of correct predictions on test data set when training entire model
  - Isn't always good
  - Can result in overfitting

When test is part of training

- Out-of-Sample Accuracy: We want this to be high

- Train/Test split: Mutually exclusive

- Determining accuracy of K-fold cross-validation:
  Average of all folds $\frac{1}{n}\sum_{i=1}^{n} a_i$ n=number a=accuracy

- No overlap in samples in this approach

Errors

- Error: Measure how far data is from fitted regression line

- Mean Absolute Error (MAE): Mean of absolute values of error

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

- Mean Squared Error (MSE): Focuses on large errors

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Root mean squared error = Interpretable in the same unit as the response vector or y units making it easy to relate its info

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_i - \hat{y}_i)^2}$$

- Relative absolute error (RAE) = Take total absolute error & normalises

$$RAE = \frac{\sum_{j=1}^{n} |y_j - \hat{y}_j|}{\sum_{j=1}^{n} |y_j - \bar{y}|}$$

MAE
Absolute error of simple predictor

- Relative squared error

$$RSE = \frac{\sum_{j=1}^{n} (y_j - \hat{y}_j)^2}{\sum_{j=1}^{n} (y_j - \bar{y})^2}$$

- $R^2 = 1 - RSE$ How closely related the data values are to the fitted line
  ↑R^2 the better

Multiple Linear Regression

Examples
- Independent variables effectiveness on prediction
  Predicting impact of changes e.g. How much does blood pressure raise with BMI rise?

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \ldots + \theta_n x_n$$

$$\hat{y} = \theta^T X$$

Equation of a line

$$X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \ldots \end{bmatrix}$$ ← Turns θ_0 into intercept or bias parameter

Goal

- Find best hyperplane for our data!
- How to estimate θ?
  - Ordinary Least Squares: Estimates θ by minimizing MSE
    - Linear algebra operations
    - Takes a long time for large datasets (10K+ rows)
  - An optimization algorithm: Gradient Descent
    - Proper approach for large datasets
- There must linear relationships between dependent and independent variables.
  - To check, we can use scatter plots & visually compare

Non-linear regression

- A polynomial regression model can be transformed into linear regression model

e.g By defining $x_1 = x$, $x_2 = x^2$, $x_3 = x^3$

$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

→ Multiple linear regression

→ Least Squares

- For non-linear regression
  - To model a non-linear relationship between dependent variable & a set of independent variables
  - $\hat{y}$ must be a non-linear function of parameters θ not necessarily features x
  - Could be exponential, logarithmic, logistic e.t.c

- How can I know if a problem is linear or non-linear in an easy way?
  - → Inspect Visually
    - Plot bivariate plots of output variables with each input variable
  - → Calculate correlation coefficient between dependent & independent variables
    - $\geq 0.7 \approx$ Linear tendency!
  - → Based on accuracy
    - If we can't accurately predict using a linear model - use non linear

Week 3 - Classification

- A supervised learning approach
- Target attribute is a categorical variable
- E.g Spam e-mail classification
- Algos: Decision trees, Naive Bayes, Linear Discriminant Analysis
  K-Nearest Neighbor, Logistic Regression, Neural Networks, SVM

K-Nearest Neighbor

- Based on their similarity to other cases with same class labels near
- Cases that are near each other are said to be "neighbors"
- Algorithm:
  1. Pick value of K
  2. Calculate distance of unknown case from all cases
  3. Select K-observations in the training data that are nearest to unknown data point
  4. Predict the response of the unknown data point using the most popular response value from the K-nearest neighbors

$$\text{Dis}(x_1, x_2) = \sqrt{\sum_{i=0}^{n} (x_{1i} - x_{2i})^2}$$

$$\text{e.g} = \sqrt{(54-50)^2 + (190-200)^2} = 10.77$$

- KNN can also be used for regression e.g median of K neighbors

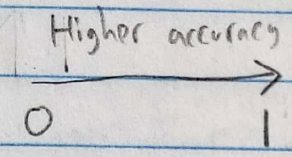
- Data standardization is good practice as KNN is based on distance

Evaluation Metrics in Classification

- Jaccard index

y: Actual labels
$\hat{y}$: Predicted labels

$$J(y, \hat{y}) = \frac{|y \cap \hat{y}|}{|y \cup \hat{y}|} = \frac{|y \cap \hat{y}|}{|y| + |\hat{y}| - |y \cap \hat{y}|}$$



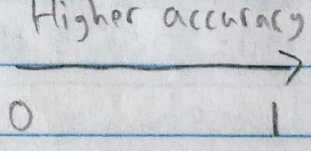
- F1-score & Confusion Matrices

TP	FN
FP	TN

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad \text{TPR}$$

$$\text{F1 score} = \frac{2\,(\text{Prc} * \text{Rec})}{(\text{Prc} + \text{Rec})}$$



- Log loss : Performance of a classifier where the predicted output is a probability value between 0 and 1

		actual (y)	Predicted ($\hat{y}$)	
e.g	1.	1	0.13	≈ Bad. High log loss
	2.	0	0.04	≈ Good. Low log loss

$$= -\frac{1}{n}\sum\left(y \times \log(\hat{y}) + (1-y) \times \log(1-\hat{y})\right)$$

log loss 1. 2.04 2. 0.04

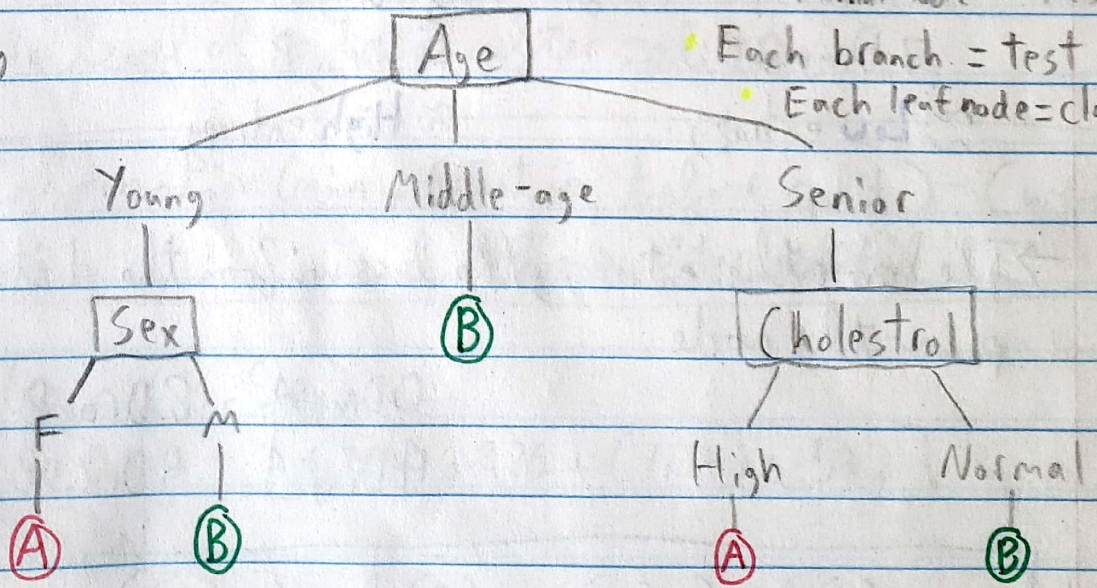
Higher accuracy →

1 ——— 0

Decision Tree

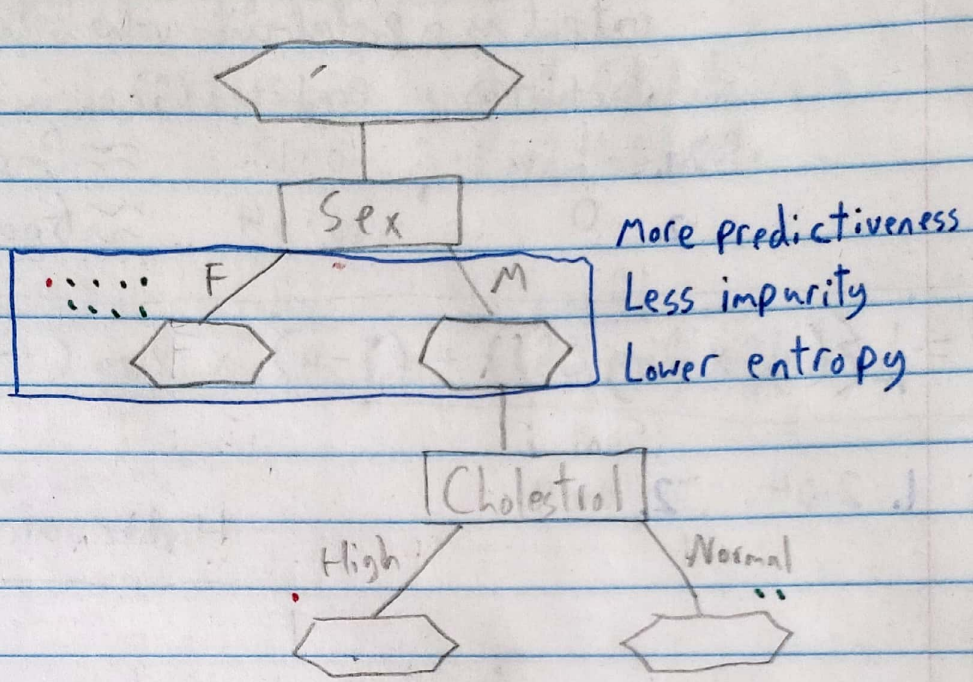
- Each internal node = test
- Each branch = test result
- Each leaf node = classification

e.g



Algo :
1. Choose an attribute from your dataset
2. Calculate attribute significance in splitting of data
3. Split data based on the value of the best attribute
4. Go to step 1

Which attribute is the best?



Entropy: Measure of randomness or uncertainty

1 Drug A	3 Drug A
7 Drug B	5 Drug B
≈ Low entropy	≈ High entropy

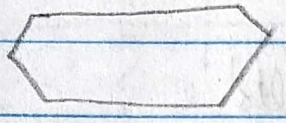
→ The lower the entropy, the less uniform the distribution, the purer the node

0 Drug A	8 Drug B	entropy = 0
4 Drug A	4 Drug B	entropy = 1

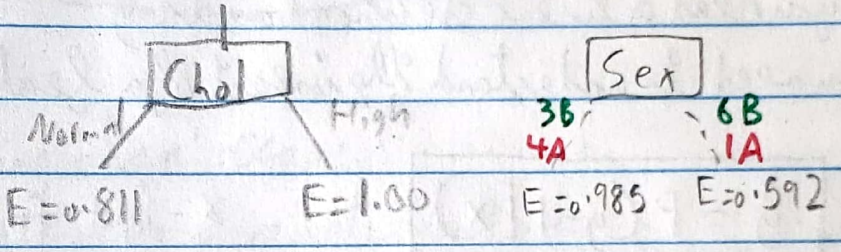
$$= -p(A)\log(p(A)) - p(B)\log(p(B))$$

Which attribute is the best one to use?

S: 9B, 5A
E = 0.940



Now, test attributes!



The tree with the higher Information Gain after splitting

Information Gain is the information that can increase the level of certainty after splitting.

Information Gain = (Entropy before split) - (weighted entropy after split)

Gain (S, Sex)
= 0.940 - [(7/14) 0.985 + (7/14) 0.592]
= 0.151

Gain (S, Chol)
= 0.940 - [(8/14) 0.811 + (6/14) 1.0]
= 0.048

0.151 > 0.048
∴ Sex is better!

Logistic Regression

- Classification algorithm for categorical variables
- Suitable when
  - If data is binary
  - If you need probabilistic results
  - When you need a linear decision boundary
  - If you need to understand the impact of a feature

$$\hat{y} = P(y=1|X)$$

X = input features

- Sigmoid function $\sigma(\theta^T X) = \frac{1}{1+e^{-\theta^T X}}$

$$\sigma(\theta^T X) \rightarrow P(y=1|X)$$
$$1 - \sigma(\theta^T X) \rightarrow P(y=0|X)$$

- Training Process:
  1. Initialize θ
  2. Calculate $\hat{y} = \sigma(\theta^T X)$ for a customer
  3. Compare $\hat{y}$ with y and record error
  4. Calculate error for all customers
  5. Change the θ to reduce the cost
  6. Go back to step 2

$\theta = [-1, 2]$

$\hat{y} = \sigma([-1,2] \times [3,5]) = 0$

error = 1 - 0.7 = 0.3

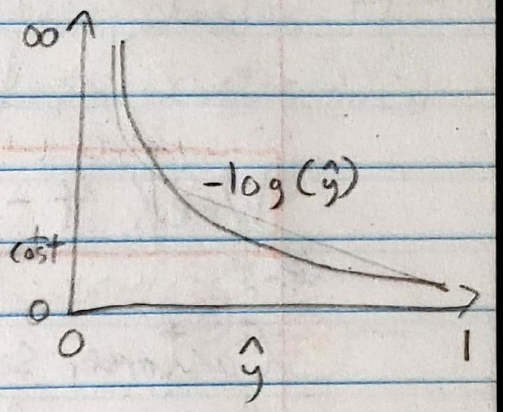
cost = J(θ)

θ_{new}

$cost(\hat{y}, y) = \frac{1}{2}(\sigma(\theta^T x) - y)^2$

$-\log(\hat{y})$ if $y=1$
$-\log(1-\hat{y})$ if $y=0$

If $Y=1$ & $\hat{y}=1$ → cost = 0
If $Y=1$ & $\hat{y}=0$ → cost = large



$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^i \log(\hat{y}^i) + (1-y^i)\log(1-\hat{y}^i)$$

Training algo recap:
1. Initialize parameters randomly
2. Feed cost function with training set, & calc error — $J(\theta)$
3. Calc gradient of cost function — ΔJ
4. Update weights with new values — θ_{new}
5. Go to step 2 until cost is small enough

SVM

Supervised algorithm that classifies cases by finding a separator
1. Mapping data to a high-dimensional feature space
2. Finding a separator

→ Kernel // From non-linear to linear separability

Adv
- Accurate in high-dim space
- Memory efficient

Disadv
- Prone to overfitting
- Only efficient with small datasets
- No probability estimation

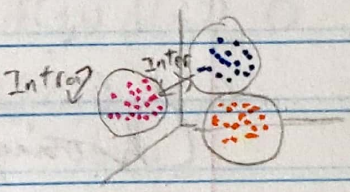
Applications: Image Recognition, Text category assignment, detecting spam, sentiment analysis, gene expression classification regression & clustering

Week 4 - Clustering

- Customer segmentation one of the most popular uses of clustering
- A group of objects that are similar to other objects in the cluster, and dissimilar to data points in other clusters.
- Unsupervised
- Clustering applications
  - Retail / Marketing
    - Identifying buying patterns of customers
    - Recommending new books or movies to new customers
  - Banking
    - Fraud detection in credit card use
    - Identifying clusters of customers
  - Insurance
    - Fraud detection in claims analysis
    - Insurance risk of customers
- Why Clustering?
  - Exploratory data analysis
  - Summary generation
  - Outlier detection
  - Finding duplicates
  - Pre-processing step

- Partitioned Based Clustering e.g. K-Means
- Hierarchical Clustering - Produces trees of clusters
- Density-based Clustering - Arbitrary shaped clusters

K-Means Clustering



- Partitioning Clustering
- K-means divides the data into non-overlapping subsets (clusters) without any cluster-internal structure
- Examples within a cluster are very
- Examples across different clusters are very different
- K-Means clustering tries to minimize the intra-cluster distances, and maximize the inter-cluster distances

$$Dis(x_1, x_2) = \sqrt{\sum_{i=0}^{n} (x_{1i} - x_{2i})^2}$$ Eucleadan dist

Can use other similarities measures such as cosine etc
-depends on domain & data types

Iterative Algo

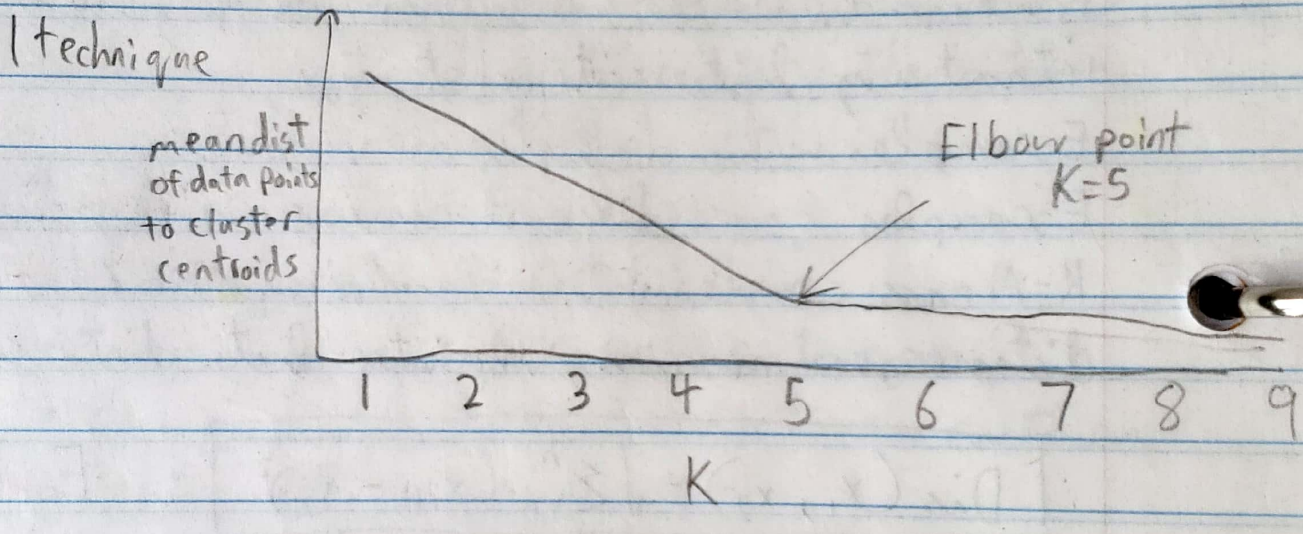
- Algorithm: Can repeat this multiple times

1. Initialize K (e.g. = 3) centroids randomly
2. Distance calculation of each data points to centroids
3. Assign each point to the closest centroid
4. Compute new centroids for each cluster based on mean of its cluster members
5. Repeat until there are no more changes

Accuracy:

- External approach: Compare the clusters with the ground truth, if it is available
- Internal approach: Average the distance between data points within a cluster

Choosing K? The elbow method//

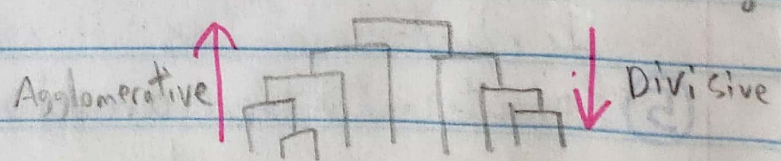


Elbow point: Where the rate of decrease sharply shifts

- Relatively efficient on medium & large datasets

Hierarchical Clustering

- Hierarchical clustering algorithms build a hierarchy of clusters where each node is a cluster consists of the clusters of its daughter nodes.



Agglomerative algorithm w/ n data points

1. Create n clusters, one for each data point
2. Compute the Proximity Matrix
3. Repeat
   i. Merge the 2 closest clusters
   ii. Update the proximity matrix
4. Until only a single cluster remains

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Distance between clusters

- Single-Linkage Clustering: Min distance between clusters
- Complete-Linkage Clustering: Max distance between clusters
- Average-Linkage Clustering: Avg distance between clusters
- Centroid-Linkage Clustering: Distance between cluster centroids

Hierarchical advantages vs disadvantages

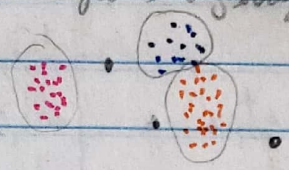
- No required # of clusters
- Easy to implement
- Produces a dendogram, helps understanding data

- Can't undo prev steps via algorithm
- Generally has long runtimes
- Can be difficult to identify the number of clusters in the dendrogram

K-means	Hierarchical
1. Much more efficient	1. Can be slow for large databases
2. # of clusters need to be specified	2. Doesn't require # of clusters to run
3. Gives only 1 partitioning of the data based on #	3. Gives more than 1 partitions based on results
4. Potentially returns different clusters each time it is run due to random initializations	4. Always generates same clusters

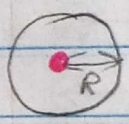
DBScan Clustering (Arbitrary-shape clusters)

- Locates regions of high density and separates outliers

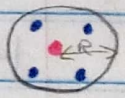


- Density-Based Spatial Clustering of Applications with Noise
  - Is one of the most common clustering algorithms
  - Works based on density of objects

- R (Radius of neighborhood)
  - Radius (R) that if includes enough number of points within, we call it a dense area

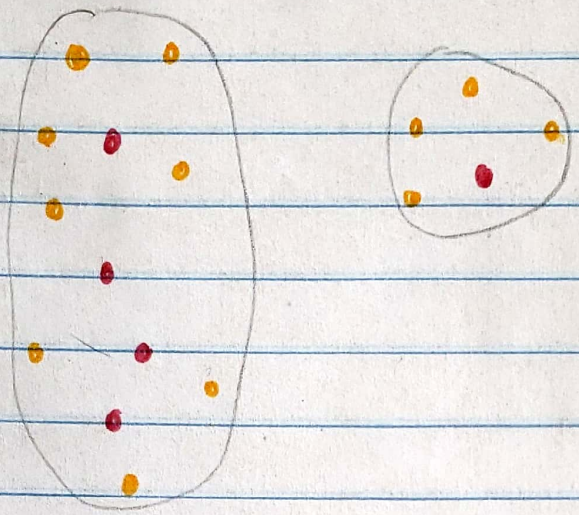


- M (Min number of neighbors)
  - The minimum number of data points we want in a neighborhood to define a cluster



- Core point (If M min around it)
- Border point (If <M min around it)
- Outlier point

Clusters? At least one core point, it's reachable core points and all their border points



- Robust to outliers
- Does not require specification of the number of clusters

Bayesian Optimization

Machine Learning Mastery

- Bayesian optimization works by building a probabilistic model of the objective model, called the surrogate function, that is then searched efficiently with an acquisition function, before candidate samples are chosen for evaluation on the real objective function.
- Often used in applied machine learning to tune the hyperparameters of a given well-performing model on a validation dataset.
- Bayesian Optimization provides a probabilistically principled method for global optimization.

→ Function Optimization Challenges ←

- Samples are drawn from the domain & evaluated by the objective function to give a score or cost
  - Samples: 1 example from the domain, represented as a vector.
  - Search Space: Extent of the domain from which samples can be drawn.
  - Obj function: Function that takes a samples and returns a cost
  - Cost: Numeric score for a sample calculated via obj function
- Summary of optimization in ML
  - Algorithm Training: Optimization of model parameters
  - Algorithm Tuning: Optimization of model hyperparameters
  - Predictive Modeling: Optimization of data + data Preparation & algo selection
- Many methods exist for function optimization
  - Random search: Randomly sampling the viable search space
  - Grid search: Systematically evaluating samples in a grid across search space

→ What is Bayesian Optimization ←

- Approach that uses Bayes Theorem to direct the search in order to find the min or max of an obj function.
- Bayes Theorem Simplified

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Posterior = Likelihood * Prior / Evidence

- Can devise specific samples $(x_1, x_2, \ldots, x_n)$ & evaluate them using the objective function $f(x_i)$ that returns cost or outcome for the sample x_i.
  → Samples & outcome collated sequentially, $D = \{x_i, f(x_i), \ldots, x_n, f(x_n)\}$

$$P(f|D) = P(D|f) * P(f)$$

  → Represents everything we know about the obj function
  → Approximation of the obj function & can be used to estimate the cost of different candidate samples that we may want to evaluate
  ∴ Surrogate obj function
  → Captures updated beliefs about the unknown obj function

- Surrogate Function: Bayesian approximation of the obj function that can be sampled efficiently
  → Sampling involves careful use of the posterior in a function known as the "acquisition" function e.g. for acquiring more samples
  → We want to use our beliefs about the objective function to sample the area of search space that is most likely to pay off.
  ∴ Acquisition optimizes conditional probability of location in search to generate next sample
- Acquisition Function: Technique by which the posterior is used to

select the next sample from search space

→ Once additional samples and their evaluation via the objective function f() have been collected, they are added to data D and the posterior is then updated.

→ The process is repeated until the extrema of the obj function is located, a good enough result is located, or resources are exhausted

u(x|data) → {(x1,y1)(x2,y2)...}

maximize x_t

Summary:

1. Select a sample by optimizing acquisition function.
2. Evaluate the sample with the objective function.
3. Update the Data and, in turn, the Surrogate function.
4. Go to 1

- Surrogate Function: Popular to treat the problem as a regression predictive modeling problem, with the data representing the input and the score representing the output to the model.
  - → Often best modeled using a random forest or a gaussian process
  - → A Gaussian Process, or GP, is a model that constructs a joint probability distribution

- Acquisition Function:
  1. Search strategy used to navigate the domain in response to surrogate function
  2. Acquisition function used to interpret and score the response from the surrogate function

UBC Youtube vid extremely useful
(Saved in Machine Learning List)