W1 Module 1 – Introduction to Analytics:

Analytics can answer:

- Descriptive questions: What happened? What factors are most important?
- Predictive questions: What will happen?
- Prescriptive questions: What action will be best?

Modelling:

1. Describe a real-life situation in terms of Math.
2. Analyze the math
3. Turn mathematical answer into real life solution.

W1 Module 2 – Classification:

Data Definitions:

- Row: Data point (single observation)
- Columns:
  - Attributes / features / predictors / covariates (a continuous variable that is expected to change ("vary") with ("co") the outcome)
  - Response / Outcome (The answer for each data point)

Data types:

- Structured data (can be stored in a structured way):
  - Quantitative (numbers with a meaning)
  - Categorical (e.g. zip code, M/F, etc.) A binary attribute (can take only 2 values) is a type of categorical data but can sometimes be used as a quantitative measure
  - Types of structured data:
    - Unrelated data (no relationship between points)
    - Time series data (same data recorded over time)
- Un-structured data (ex. Text)

Classification the process of predicting the class/category of given data points (putting data points into categories based on similarity). Example of Models:

- Support Vector Machines (SVM)
- K-Nearest Neighbor (KNN)

Support Vector Machines:

n <- number of data points

m <- number of attributes

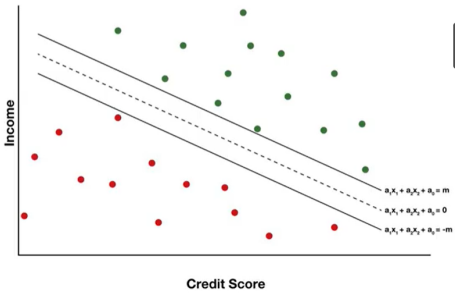
x_{ij} is the i attribute of the j data point

y_j is the response of the data point j

- $y_j = 1$ if the data point is green
- $y_j = -1$ if the data point is red

line $a_1x_1 + a_2x_2 + a_3x_3 \dots + a_mx_m + a_0 = 0$

$$\sum_{i=1}^{m} a_ix_i + a_0 = 0$$



Green points: $a_1x_{1j} + a_2x_{2j} + \dots + a_mx_{mj} + a_0 \geq 1$

All points: $(a_1x_{1j} + a_2x_{2j} + \dots + a_mx_{mj} + a_0)y_j \geq 1$

Red points: $a_1x_{1j} + a_2x_{2j} + \dots + a_mx_{mj} + a_0 \leq -1$

Distance between 2 lines: $\frac{2}{\sqrt{\sum_{i=1}^{m}(a_i)^2}}$ (not including a0)

Objective is to maximize distance (margin), thus minimize $\sum_{i=1}^{m}(a_i)^2$ called margin denominator (hard separation)

Soft classifiers are used when there is no hard boundary between data points and their classes.

Error for data point j : $\max\{0,1 - (\sum_{i=1}^{m} a_ix_{ij} + a_0)y_j\}$

Objective: Minimize total error for all data points $Minimize \sum_{j=1}^{n} \max\{0,1 - (\sum_{i=1}^{m} a_ix_{ij} + a_0)y_j\}$

SVM Model Equation:

$Minimize$ (Total Error + λ Margin)

$\sum_{j=1}^{n} \max\{0,1 - (\sum_{i=1}^{m} a_ix_{ij} + a_0)y_j\} + \lambda \sum_{i=1}^{m}(a_i)^2$

As $\lambda \to 0$, the importance of minimizing mistakes (errors) in classifying data points outweighs margin.

Extension of SVM:

How to handle Cost of misclassification:

For Hard classification: Adjust the intercept (a_0 can range from -1 to 1 without misclassification)

For soft classification (add point weights):

$Minimize$ (m_j * Total Error + δ Margin)

$\sum_{j=1}^{n} m_j * \max\{0,1 - (\sum_{i=1}^{m} a_ix_{ij} + a_0)y_j\} + \delta \sum_{i=1}^{m}(a_i)^2$

$m_j > 1$ for costlier errors

$m_j < 1$ for less costly errors

After scaling the data, the value of the coefficients a_i indicates the importance of the attribute.

($a_i \to 0$) attribute not important for the model.

SVM works for:

- As many dimensions as needed
- does not have to be linear models, Kernel methods allow for non-linear SVM models.

Adjusting the data (Data preparation):

Scaling linearly:

- data between 0 and 1 ($\frac{x - x_{min}}{x_{max} - x_{min}}$)
- data between a and b ($\frac{x - x_{min}}{x_{max} - x_{min}}$) (b-a) + a

Standardization:

- Scaling to a normal distribution
- Common scaling: $\mu = 0, \sigma = 1$
- $\frac{x - \mu}{\sigma}$

K-Nearest Neighbor (KNN):

Predict the class of a new data point.

- Find the k-closest points
- The predicted class is the most common (mode) in the neighbors.

Notes:

- Can use different distance metrics (norm)
- Attributes can be weighed by importance
- Irrelevant attributes can be removed

W2 Module 3 – Validation:

Validation: How accurate the model is.

Data has 2 patterns:

- Real effects: actual relationships between attribute and response. Same in all datasets
- Random effects: different in all datasets

Testing on training data makes model looks better. It fits both real and random effects.

Splitting the data:

- Training: larger dataset to build the model.
- Validation: Smaller dataset to compare different models and choose the best model. (Note best model could have benefited more from random effects)
- Testing: Smaller dataset to estimate the actual performance of the final model.

Rules of thumb:

- 1 Model: 70-90% training, 10-30% testing
- Comparing models: 50-70% training, split equally between validation and test.

Data splitting approaches:

1. Random
2. Rotation (risk: can introduce more bias)

Both types need to be checked for bias.

K-Fold Cross-validation (Common k = 10):

- Split (training + validation) into K datasets
- For each of the k parts, train the model on all the other parts then evaluate it on the k part.
- Average all results to estimate model quality.

After model selection, Build the final model using (training + validation) datasets.

Importance of cross-validation:

- Better use of data

- Better estimate of model quality Which leads to Choosing model more effectively

W2 Module 4 – Clustering:

Clustering (Response unknown) helps in:

- Grouping the data (e.g. marketing)
- Discovering what groups are in the dataset (analyze each group independently)

P-norm Distance:

- Euclidian distance (straight line) (2-norm)
  $d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$
- Rectilinear distance (1-norm)
  $d = |x_1 - y_1| + |x_2 - y_2|$
- General formula (p-norm)
  $$d = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}$$
- Infinity norm (largest absolute of a set of numbers)
  $d = \max(|x_i - y_i|)$

K-Means algorithm:

Find optimum location of cluster centers to minimize the total distance between the points and cluster centers. Methodology:

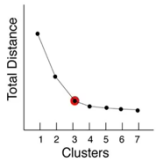
1. Pick k cluster centers within range of data
2. Assign each point to nearest cluster center
3. Recalculate cluster centers
4. Repeat steps 2 & 3 until no changes.

K means algorithm is:

1. a machine learning algorithm
2. a heuristic method (fast, good but not guaranteed to find best solution)
3. expectation maximization model (EM)

Find optimum number of clusters (k):

- Qualitative (number of police stations in budget)
- *Elbow diagram*



Note: Total distance from point to cluster centers

K-means prediction:

- For a new point, the nearest cluster center
- For many points (or to find range of clusters), Voronoi diagram

Supervised learning: response is known (e.g. classification)
Unsupervised learning: response is not known (e.g. clustering)

W3 Module 5 – Basic Data Preparation:

Data preparation:

1. Dealing with Outliers (Step 1)
2. Scaling / standardization of the data.

Outlier is a point(s) that's very different from the rest.

Types of outliers:

- Point outliers
- Contextual outliers: a value that isn't far from the rest overall, but far from points nearby in time. (time series).
- Collective outlier: something is missing in the range of data (outlier by omission)

Finding outliers for 1 dimension:

- Automated methods: Box and Whisker plot / QQ-Plot
- Grubbs test (hypothesis testing)

For multi-dimensions or contextual/collective outliers:

- Modelling error: build model and find major errors points

Dealing with outliers (Always needs investigation):

- Bad data:
  - Omit
  - Data imputation
- Real data:
  - In large datasets, outliers are normal.
  - Removing outliers could lead model to be optimistic.

Alternative: build a model to estimate probability of outlier happening.
Sometimes outliers need to be removed even if real.

W3 Module 6 – Change Detection:

Detecting a change in a time series.

Objective:

- Define whether an action is needed.
- Determine impact of past actions.
- Determine changes to help plan (predict or avoid future problems).

Cumulative Sum (CUSUM) technique

Detect: increase / decrease or both.

Increase:

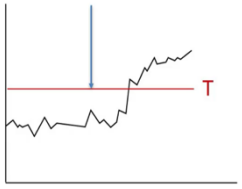
$$S_t = \max(0, S_{t-1} + (x_t - \mu - C))$$
$$S_t \geq T\ ?\ (Threshold)$$

Decrease:

$$S_t = \max(0, S_{t-1} + (\mu - x_t - C))$$
$$S_t \geq T\ ?\ (Threshold)$$

- Increasing Critical value and/or threshold, decreasing model sensitivity
- Choosing C & T optimum values is a tradeoff between the cost of late detection versus false detection.

Control chart (plot S_t versus time)



W4 Module 7 – Time Series Models:

Variation in time series data:

- Trend changes: increase or decrease over t
- Cyclical changes: cycles (daily, weekly etc.)
- Random variations

Exponential smoothing method:

S_t is expected response at time t (baseline)
x_t is actual/observed response at time t

$$S_t = \alpha x_t + (1 - \alpha) S_{t-1}$$
$$0 < \alpha < 1$$

$\alpha \to 0$ a lot of randomness in the system (more weight to previous baseline)
$\alpha \to 1$ small randomness in the system (more weight to current response)
Initial condition $S_1 = x_1$

Adding trends (Double smoothing):

T_t is the trend at time t

$$S_t = \alpha x_t + (1 - \alpha)(S_{t-1} + T_{t-1})$$
$$T_t = \beta(S_t - S_{t-1}) + (1 - \beta) T_{t-1}$$
$$0 < \beta < 1$$

Initial condition $T_1 = 0$
T > 0, increasing trend.

Adding Cyclical changes (Triple smoothing, Holt-winters method):

Seasonality: Multiplicative cyclical changes
L the length of the cycle
C_t the multiplicative seasonality factor

$$S_t = \frac{\alpha x_t}{C_{t-L}} + (1 - \alpha)(S_{t-1} + T_{t-1})$$
$$C_t = \gamma \frac{x_t}{S_t} + (1 - \gamma) C_{t-L}$$
$$0 < \gamma < 1$$

Note C_{t-L} cyclical factor from last cycle (t-L)
Initial Condition $C_1 \ldots . C_L = 1$ (first L factors of C)
If C_t = 1.1, 10% increase due to cyclic effects.
Find optimum γ, β and γ is based on optimization for minimizing error.

Exponential smoothing –Name origin:

- Smooth out the data.
- Exponential because every past observation contributes to the current baseline estimate and newer observations are weighed more.

Exponential smoothing can be used for simple short term forecasting.

- Best estimate of baseline is current baseline
  $$F_{t+1} = S_t$$
- Best estimate of trend is current trend
  $$F_{t+1} = (S_t + T_t) * C_{(t+1)-L}$$

Auto-Regressive Integrated Moving Average (ARIMA):

A more general method for analyzing time series.

ARIMA combines:

- Differences: uses the difference rather than actual observed data. (1st degree difference, etc. called d^{th} order difference). Exponential smoothing requires the data to be stationary (mean and variance constant over time).
- Auto regression: Using p time periods back instead of all the time periods (called infinity order) in exponential smoothing. Use older values of the same variable to predict.
- Moving Average: Uses previous error for q time periods as predictors (order q moving average).
- Can add seasonality.

Given p, d and q, the model can be built for other constants using optimization.
ARIMA (0,1,1) exponential smoothing model
Needs minimum 40 data points.

Advantages:

1. A more general model

2. Can be used for short term forecasting
3. Better than exponential smoothing if the data is more smooth.

GARCH Models:

- Used to estimate / forecast the variance (squared errors) not value of observations (how much higher or lower the forecast could be from the true value, variance as a measure of risk).
- Does not deal with difference of variance (only raw variance)

W5 Module 8 – Basic Regression:

Regressions answers 2 types of questions:

- Descriptive: How systems work? Key factors? E.g. effect of economic factors on elections
- Predictive: What will happen in the future? E.g. forecast oil price in the future
- Cannot answer prescriptive questions

Simple linear regression (SLR):

$$y = a_1 x_1 + a_0$$

Multi linear regression (MLR):

$$y = \sum_{i=1}^{m} a_i x_i + a_0$$

Measures of the quality of the model:

1. Sum of squared errors (SSE):

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

$$SSE = \sum_{i=1}^{n} (y_i - (a_0 + \sum_{j=1}^{m} a_i x_{ij}))^2$$

   - Based on maximum likelihood MLE approach.
   - Assuming independent normally distributed errors with mean 0, then the set of model parameters that minimizes SSE is the maximum likelihood fit.

2. Akaike Information Criterion (AIC):
   - Used to compare different models. (Smaller AIC are better models)
   - Combines MLE with model simplicity
   - $AIC = 2k - 2\ln(L^*)$
   - Where k is the number of parameters in the model & L* is the maximum likelihood value
   - Has a penalty term (2k) to balance likelihood with model simplicity (avoid over-fitting)
   - Good if there are infinitely many data points (Thus it needs a correction term)
   - Corrected AIC

$$AIC_c = AIC + \frac{2k(k+1)}{n-k-1}$$

   - Where n is the of observations.
   - Can calculate the probability of a model being better than another

$$relative\ likelihood = e^{(AIC_1 - AIC_2)/2}$$

   - Example: Model 1 AIC = 75 & Model 2 = 80.
   - $relative\ likelihood = e^{(75-80)/2} = 8.2\%$
   - Model 2 is 8.2% as likely as Model 1 to be better (Model 1 is probably better)

3. Bayesian Information Criterion (BIC):
   - $BIC = k\ln(n) - 2\ln(L^*)$
   - Penalty term BIC > AIC (encourages simpler models)
   - Only use BIC when number of data points (n) a lot more than parameters
   - Rules of thumb
     - $|BIC_1 - BIC_2| > 10$
       - smaller-BIC model is "very likely" better
     - $6 < |BIC_1 - BIC_2| < 10$
       - smaller-BIC model is "likely" better
     - $2 < |BIC_1 - BIC_2| < 6$
       - smaller-BIC model is "somewhat likely" better
     - $0 < |BIC_1 - BIC_2| < 2$
       - smaller-BIC model is "slightly likely" better

Causation vs Correlation:

Regression models shows correlation not causation.
Correlation does not mean causation.
When there is a causation:

- Cause before effect
- Idea of causation makes sense (open to disagreement)
- No outside factors causing the relationship (hard to guarantee or prove)

Transforming the data:

- We can use any quadratic function in regression.
- Always keep original variables (hierarchy)
- We can use variable interaction (combine 2 or more attributes as an interaction term).
- We can transform (Box-Cox, log, etc.) predictors or interaction terms or response or all of them.

Linear Regression outputs:

p-value: of each coefficient (p<0.05 good predictor), a type of hypothesis testing (probability of coefficient being equal to 0).

- Higher threshold: more factors included, higher change of including irrelevant factors.

3 Warnings

- P-values gets smaller with very large data points even if the attribute is irrelevant.
- P-values are only probabilities. 0.02 p-value means 2% chance of being irrelevant
- P-value for a predictor are model specific.

Confidence interval (CI): where the coefficient probably lies (typical 95% confidence) & how close it is to 0 or if it overlaps 0.
T- statistic: coefficient divided by standard error (similar to p-value)
Value of coefficient itself: Could be no change when multiplied by the attribute value (even if it has a low p-value).
R-squared (R2): measure of how much variability the model accounts for.
Remaining is either randomness or other factors not accounted for.
Adjusted R2 adjusts for the number of attributes used.

W6 Module 9 – Advanced Data Preparation:

Why do we need data transformation?

- Some models assume data is normally distributed. (if data is not, causes model bias)

Heteroscedasticity: unequal variance in different ranges of the data

Box-Cox transformation:

- logarithmic transformation
  - stretches out the smaller range to increase its variability.
  - shrinks the larger range to decrease its variability.

$$t(y) = (y^{\lambda} - 1)/\lambda$$

Objective: find optimum lambda.
Check the need for transformation using Q-Q plot

De-trending data:

- Used to remove the trend from time series data
- Applies to response / predictor
- For factor-based models (regression, SVM, etc.) to avoid trend impacting the model results.

How to detrend?

- By other known data, e.g. inflation rate
- Factor by Factor using 1D regression fit $y = a_0 + a_1 x$ (trend fit)
- Note could use more complex regression fits

de-trended data = Actual value – trend fit

Principal Component Analysis (PCA):

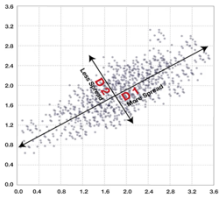
- A Feature extraction technique
- Used for high dimension & correlated data (lots of factors)

Importance of PCA:

- Reduce the number of predictors (ranks coordinates by importance)
- Address high correlation between predictors (changes coordinates to remove correlations within the data)

Concentrate on the first n principal components:

- Reduces the effect of randomness (Earlier principal components are likely to have higher signal to noise ratio)



D_1 is more important than D_2
ALL PCA is after scaling

Mathematics:

- X is a scaled matrix
- Find all eigenvectors of $X^T X$
- V is the matrix of eigenvectors (sorted by eigenvalues)
- PCA_1 = X*V1 (First column of V)

Original factors coefficients can be found after PCA model is done (if all PCA's were used).

Eigenvalues and Eigenvectors:
v is vector: $Av = \gamma v$
v: eigenvector of A
γ: eigenvalue of A
Given γ solve $Av = \gamma v$ to find corresponding eigenvalue v

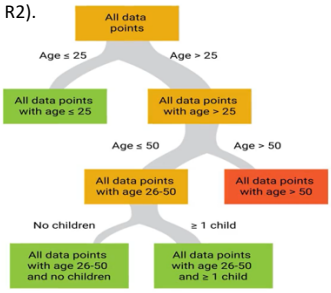
W7 Module 10 – Advanced Regression:

Classification and Regression Trees (CART):
Tree models can be used for Classification, Regression or Decision making (Decision tree)

Trees in Regression:

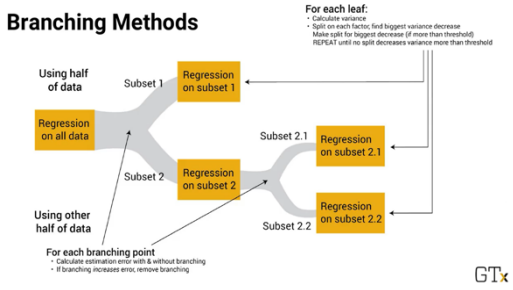
- Data points are divided into branches
- Branch endings (groups of points) are called leaves
- For each leaf, create a separate model to find the coefficient.
- As a result, it will help:

1. Descriptively, use each leaf coefficients to explain the behavior of its group.
2. Predictively,
   a. Better prediction models for each leaf and the overall model
   b. targeted prediction for specific groups.
   c. investigate possible additional predictor for low model quality leaves (low R2).



How to branch trees:

- Which factors should be in the branching decision?
  - Could be as many factors (combination of factors) as needed
  - No good algorithm
  - Common practice, use 1 factor at a time.
- Branching Method Logic (other methods possible):
  - Using a metric related to model's quality (e.g. R2)
  - Start with half of the data (QC with other half).
  - Find the best factor (and value of factor) to branch with (showing biggest variance decrease and still has minimum data points in each branch)
  - Check if model really improved above a certain threshold. If not prune the branch.
  - Rule of thumb: each leaf contains >= 5% of the original data, else prune the branch



Random Forest Method:

1. Introduce randomness (by bootstrapping: each tree has n random points which could be multiples of some points and none of others) each tree has a slightly different dataset
2. Generate many different trees (with different strengths and weaknesses)
   - For Branching:
     - Randomly select a small number of factors (not all factors as in tree model). Common number 1+ log(m)
     - Choose the best factor to branch on
   - We don't need to prone the tree.
3. Use average predicted response or (mode in classification) of all trees. Default Model (black-box predictor with no detailed insight)

Benefits	Drawbacks
• Better overall estimates • Averages between trees somewhat neutralizes over-fitting	• Harder to explain/ interpret results • Can't give us a specific regression or classification model from the data

Logistic Regression (Logit Model):
Need to generate responses between 0 and 1.
Linear Regression Model

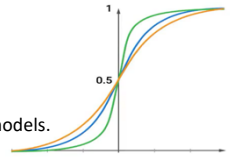
$$y = \sum_{i=1}^{m} a_i x_i + a_0$$

Logistic Regression Model

$$\log \frac{p}{1-p} = \sum_{i=1}^{m} a_i x_i + a_0$$

$$p = \frac{1}{1 + e^{-(\sum_{i=1}^{m} a_i x_i + a_0)}}$$

If $\sum_{i=1}^{m} a_i x_i + a_0 = -\infty$ then p=0
If $\sum_{i=1}^{m} a_i x_i + a_0 = +\infty$ then p=1
Logistic regression curve shape. Coefficients govern the shape of the logistic regression curve.



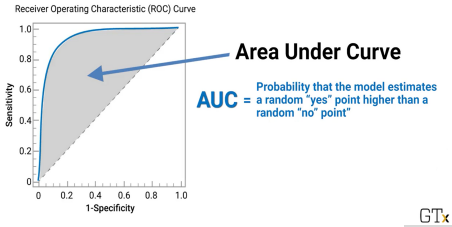
Similarities to linear regression, we can:

- Transform input data
- Consider interaction terms
- Use variable selection methods
- Logistic regression trees and random forest models.

Differences from linear regression:

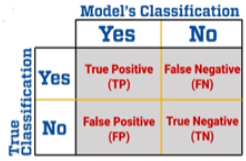
- Longer to calculate (No closed form solution)
- Difficult to understand model quality (needs pseudo R2 which is not really measuring fraction of variance explained by the model)

Logistic regression can be used in classification by setting probability threshold at some value say >0.7 for 1 else 0.



ROC/AUC is a quick estimate of model quality but does not consider the cost of false positive or false negative. Better to use confusion matrix. (AUC = 0.5 guessing)

Confusion Matrix:
Measure how well a classification model works (SVM, Logistic regression, KNN)



Sensitivity = $\frac{TP}{TP+FN}$ (Fraction of category members correctly identified)
Specificity = $\frac{TN}{TN+FP}$ (Fraction of non-category members correctly identified)
Evaluating a model quality:

- Needs 1 more input: cost of lost productivity
- Compute cost and compare models.

Advanced Regression Models:

- Poisson regression (when response has Poisson distribution)
- Regression splines (polynomials connected to each other – fit different functions to different parts of the data set with smooth connection between parts)
- Bayesian regression (starts with estimate of coefficients and random error distribution – Based on Bayes theorem- helpful with combining expert opinion with small datasets)
- K-Nearest Neighbor regression: same as classification but averaging of response instead of mode.

W8 Module 11 – Variable Selection:

Reasons to limit number of factors in a model:

1. Avoid Overfitting
   - When # factors is close or larger than # of data points, the model might fit too closely to random effects instead of real effects.
2. Simplicity
   - Easier to explain
   - Easier to demonstrate that the model is not using any illegal factors or other factors that is highly correlated with them.
   - Less data is required (less cost of data acquisition)
   - Less chance of including insignificant factors (p-value is a probability)

Methods of Variable Selection:

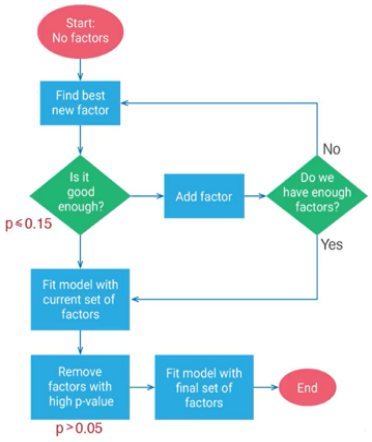
a) Greedy Algorithms
   - Criteria can be p-value/AIC/BIC/R2
   - Decisions are made step by step
   - At each step do 1 thing that looks best
   - Future options are not considered
     1. Forward Selection
     2. Backward Elimination
     3. Stepwise Regression (Most Common)

b) Global approaches
   1. Lasso Approach
   2. Elastic Net

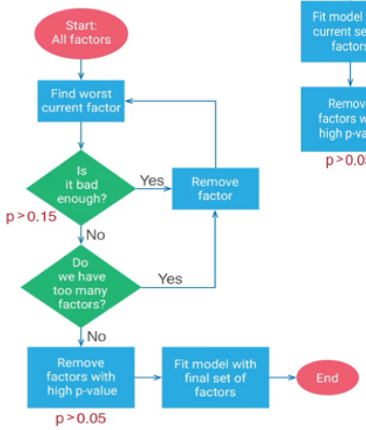
Comparison Between Different Methods

Greedy Methods	LASSO/Elastic Net
Good for initial analysis Don't perform well in prediction	Slower Better prediction

Forward Selection

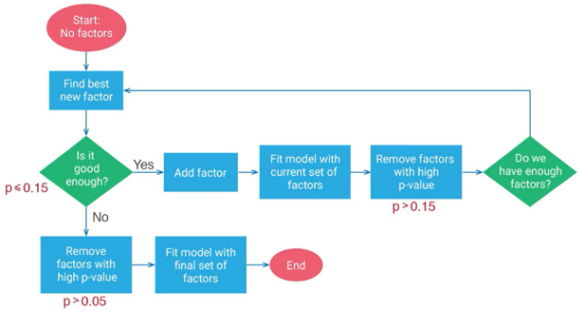


Backward Elimination



Stepwise Regression

A combination of Forward selection and backward elimination



LASSO Approach

Regression equation (minimize SSE)

$$\min\left[\sum_{i=1}^{n}\left(y_i-\left(\sum_{j=1}^{m}a_i x_{ij}+a_o\right)\right)^2\right]$$

LASSO add a constraint on the sum of all coefficients (After scaling the data)

$$\sum_{i=1}^{j}|a_i|\le t$$

Value t depends on:

- Number of variables needed
- Quality of the model with more variables

Best approach: try different t and find best tradeoff between model quality and number of variables

Elastic Net Approach

Similar to LASSO Approach but add a constraint on the sum of all coefficients and their squares (After scaling the data)

$$\lambda\sum_{i=1}^{j}|a_i|+(1-\lambda)\sum_{i=1}^{j}a_i{}^2\le t$$

Same method to choose t as LASSO.

Ridge Regression:

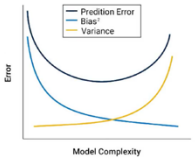
Elastic Net = LASSO + Ridge Regression (Constraints)

Ridge regression will choose smaller (in an absolute sense) non-zero coefficients for both models. By nature, it may underestimate the effect of the factors.

Ridge Regression add a constraint on the sum of squares of all coefficients (After scaling the data)

$$\sum_{i=1}^{j}a_i{}^2\le t$$

Prediction Error:



Function of Both Bias and Variance

- Ridge regression pushes the coefficients towards 0 to reduce variance (regularizes the coefficients) – Better Prediction
- For LASSO some coefficients are forced to 0 to simplify the model (Bias for variable selection)

Advantages of Elastic Net:

Combines the benefits of LASSO & Ridge Regression

- Better variable selection (LASSO)
- Better Prediction (Ridge Regression)

Disadvantages of Elastic Net:

Combines the drawbacks of LASSO & Ridge Regression

- Arbitrary rules out some correlated variables (LASSO)
- Underestimates the coefficients of very predictive variables (Ridge Regression)

No rule of thumb for which approach use.

W9 Module 12 – Design of Experiments:

Why Do we need Design of Experiments (DOE)?

- DOE is used before collecting data.
- When Full data set is hard to get (e.g. Surveys).
- Estimate the importance of each factor.

Important concepts in DOE:

- Comparison and Control: Need to control other factors in the experiment.
  - Ensure datasets have same mix of factors
  - Break dataset into smaller sets with the same other factors.
- Blocking: a blocking factor creates variation. Need to account for a blocking factor variance.

A/B Testing:

- An Experiment used to compare between 2 (or more) alternatives to determine which one performs better.
- A hypothesis test is then done to compare the options performance (Common approach in marketing).

A/B Testing requirements:

1. Be able to collect data quickly
2. Data must be representative of the population

3. Amount of data small compared to full dataset

Full Factorial Design:
An Experiment used to estimate the effect of multiple factors. Test every possible combination of factors (Needs limited options to be feasible).

- Analysis of Variance (ANOVA) to determine the importance of each factor.

Fractional Factorial Design:
An Experiment used to estimate the effect of multiple factors. Test subset of combination of factors.
Balanced Design:

- Test each choice the same # of times
- Test each pair of choices the same # of times.

In case of independent Factors:

- Test subset of combinations of choices
- Use regression to estimate effects.

Multi-Armed Bandit:
Tradeoff between more information vs. immediate value. (Exploration [getting more information] vs. Exploitation [immediate value])
Example 10 alternatives with 10,000 tests:

- 1,000 shown best alternative (max. value)
- 9,000 tests lost value

Methodology of Multi-Armed Bandit:

- Assume k alternatives
- Start with no information
  - Equal probability of selecting any alternative.
- Repeat for n times.
  - Update probability of each alternative.
  - Design new test based on new probabilities
  - Repeat until best answer is clear

Parameters:

- Change # of tests between recalculating probabilities.
- Change the way we update probability (e.g. Bayesian updates or from observed distribution)
- Change the way we pick an alternative (new) test.

Advantage: Maximize value while continuing to get more information.

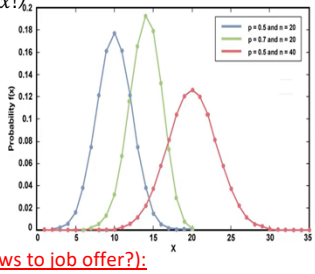
W9 Module 13 – Probability based Models – Part 1:
Probability distribution could be a simpler approach to solve a problem or form the backbone of simple models

Bernoulli Distribution (Flipping a coin, charity ask for donations):
A single event with 2 independent outcomes.
More useful when we put more together.
Example: Flipping a coin (50% chance)

$$P(X = 1) = p\ (heads)$$
$$P(X = 0) = 1 - p\ (Tails)$$

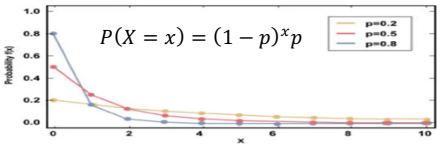
Binomial Distribution:
Probability of getting x success out of n independent identically distributed Bernoulli (p success) trials.
Large n -> Binomial converges to normal distribution

$$P(X = x) = \left(\frac{n!}{(n-x)!\,x!}\right) p^x (1-p)^{n-x}$$



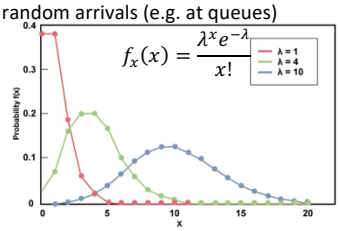
Geometric Distribution (how many interviews to job offer?):
How many Bernoulli trials until…?
Probability of having x Bernoulli (p) failures until first success. Or having x Bernoulli $(1-p)$ successes until first failure.

Note: p here is the probability of the outcome in question (good or bad)



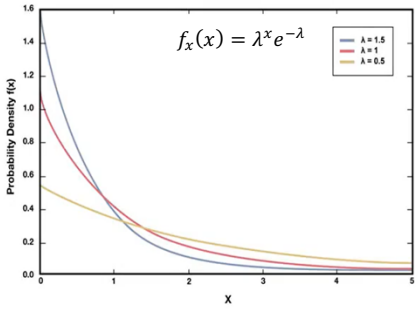
Assumption: Each Bernoulli trial is independent and identically distributed (i.i.d). We can compare data to geometric distribution to test whether i.i.d is true, else other factors are impacting the data.

Poisson Distribution (random no. people arrive at ques, no. calls to call center):
Finds the probability that x successes (p) occur given arrival rate λ (Average number of arrival / time period).
Good at modelling random arrivals (e.g. at queues)



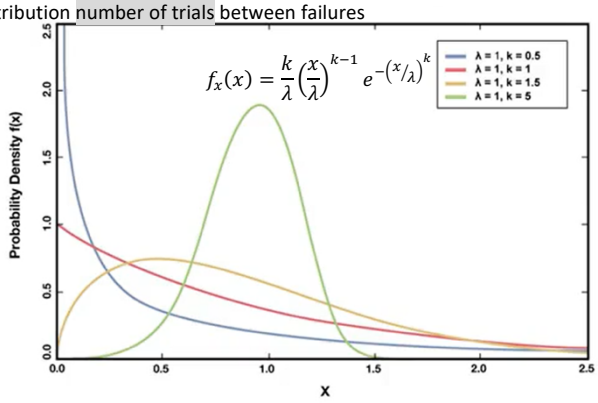
Assumption: arrivals are independent and identically distributed (i.i.d).

Exponential Distribution (time between arrivals):
If arrivals are Poisson (λ) distribution then time between arrivals is exponential (λ) distribution and vice versa.



Good at modelling time between random arrivals (e.g. at queues)

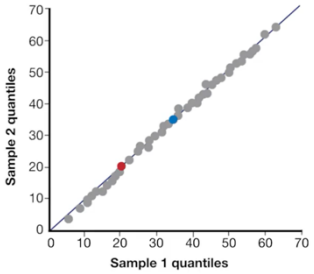
Weibull Distribution (How long the light bulb fails):
Good at modelling amount of time between failures. While Geometric distribution number of trials between failures



K<1	Modeling failure rate decreases with time. (worst things fail first) e.g. defective parts
K>1	Modeling failure rate increases with time. (Things that wear out) e.g. tires.
K=1	Constant failure rate over time (It reduces to the exponential distribution)

Q-Q Plots (Visualization, single or multiple data set):
Used to Compare visually whether

- Two datasets have the same distribution
- The dataset is distrusted following a probability distribution.



Concept: regardless of the number of points and variation 2 similar datasets should have the same value at each quantile (or percentile)
When comparing to distribution Y-axis has theoretical distribution values.

W9 Module 13 – Probability based Models – Part 2:

Queuing Models:

- Can be solved by math if simple enough else we can use simulation.
- Potential queuing parameters (Kendall Notation)
  - Arrival distribution [A]
  - Service distribution [S]
  - Number of servers [c]
  - Size of the queue [K]
  - Population size of Arrival [N]
  - Queuing Discpline [D]
- Model extension examples:
  - Hang-ups
  - Balking (when people see how many in front of them in a queue and leave)

Memory-less property:

- It doesn't matter what happened in the past, all that matters is where we are now.
- Attribute of Poisson & exponential probability distribution.
- Distribution of remaining call time = initial distribution of call time (exponential).
- Distribution of time to next arrival = initial distribution of time to next arrival (Poisson).
- If data fits Exponential or Poisson distribution, it has to be memoryless and the opposite is true.

Simulation:

- Building a model of something to study and analyze its behavior.
- Type of Prescriptive analytics: answers what if scenarios.
- Model is only as good as quality of input data (Missing or incorrect information lead to incorrect answers)

Types of simulation:

- Deterministic simulations:
  - Same inputs give same outputs (no randomness)
- Stochastic simulations:
  - Used when systems have randomness.
  - Outputs differ for the same inputs
- Continuous time simulation:
  - Continuous change over time
    - Example: chemical process, propagation of disease
  - Often modelled with differential equations
- Discrete event simulations:
  - Changes only occur when something happen
    - Example call center simulations (changes occur when a call is received or operator ends the call)

Discrete events stochastic simulations:

- Valuable when systems have high variability (using average is not good enough)
- Important to evaluate the model against real life to validate the model (variability and average)

Elements of a simulation model:

- Entities: things that move through a simulation (bags, people etc.)
- Modules: parts of a process (queues, storage)
- Resources (workers)
- Actions
- Decision points
- Statistical tracking

Replications: number of simulation runs.
Run stochastic simulations multiple times (not once) and evaluate the outcomes distribution (not only average value)
In scenarios comparison we can use the same random numbers (seed numbers) for the multiple cases of each scenario.

Simulation Use & Comparison:

- Simulation can be a powerful tool
  - Model is only as good as quality of input
  - Missing or incorrect information may lead to incorrect answers

Markov Chain:

- Probability based model
- Based on states of a system
- It's memoryless (state transitions depends only on the most recent state) – Most systems do not exhibit that.

Transition matrix

From \ To	Sunny	Cloudy	Rainy
Sunny	.75	.15	.10
Cloudy	.20	.40	.40
Rainy	.40	.30	.30

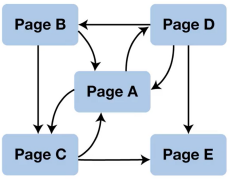
P_{ij} is the transition probability from state i to state j
How to calculate long run probability of a certain state (Rainy)?

- Assume probability of states today (initial state)
  $$\pi = (0.5, 0.25, 0.25)(Sunny, Cloudy, Rainy)$$
- Probabilities of states tomorrow =
  $$P_{S\_tmrw} = 0.5 * 0.75 + 0.25 * 0.2 + 0.25 * 0.4$$
  $$\pi P = (0.525, 0.25, 0.225)$$
- And so on new π (πP) then πP to find day after tomorrow probabilities

If we continue for ∞ days πP^{∞}, we reach steady state:

- Probabilities does not depend on initial state.
- $\pi^* P = \pi^*$ and thus π^* remains constant
- Solve for π^* and find Steady state probabilities such that $\sum \pi_i^* = 1$
- Note π^* might not exist. Assumes each state must be reachable from all other states and no cycles between states. Not true for most real world applications.

Real application: Web search example



P_{ij} is the link between pages or 0 if no link.
Use Markov chain for jumping randomly from 1 webpage to another.
Find Steady state probabilities π^* to know the probability of a random surfer landing at each page. (used to rank web pages)

W10 Module 14 – Missing Data:

Data Problems:

- Missing data
- Wrong data
- Patterns in missing or wrong data (Some data are more likely to be missing – Bias in the data)

Handling Missing data:

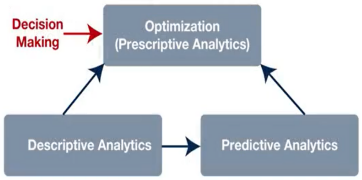
1. Removing missing data points
   - Pros:
     - Easy to implement
     - No potential for introducing errors
   - Cons:
     - Lose too many data points
     - Potential for Biased missing data (if there is a pattern for missing values)
2. Use categorical variables to indicate missing data (Non-Imputation):
   - If the variable is categorical, add a new category "Missing".
   - If the variable is quantitative,
     - Set all missing values to 0
     - Add a new categorical variable to indicate missing data
     - Add interaction variables between categorical variable and other columns to reduce bias if missing data has patterns (results in 2 models).
3. Estimate missing values (Imputation):
   - Mean, Median (numeric) or Mode (Categorical):
     - Pros:
       - Easy to implement
       - Hedge against being too wrong
     - Cons:
       - Potential for Bias (if there is a pattern for missing values)

- Regression:
  - Use model to predict missing data based on other factor
  - Pros:
    - Reduce the problem of bias if missing data has patterns
    - Usually gives better values
  - Cons:
    - Complex: Build, fit, validate, test to estimate missing values.
    - Use same data twice (to predict missing value & final model, could lead to over fitting)
    - Does not capture all variability. (Add Perturbation to each imputed value)
      - Adds variability
      - Less accurate on average

Limitation on imputation (Data used twice, overfitting):
- No more than 5% per factor
- 1 missing value per data point (other techniques are possible)

Errors from imputation:
Imputation error + perturbation error + model error

W10 Module 15 – Optimization Part 1:
A tool for Prescriptive analytics (Provide guidance on decisions).



Software automates solution but Model Building done by the analyst (No software for model building)
Components of Optimization models:
- Variables – Decisions that the software will pick the best value for.
- Constraints – Restrictions on variables values.
- Objective function – Solution quality measure. Maximize/Minimize some value. (Usually need other models to determine the input of Objective functions)

Solution: Values for each variable
- Feasible solution: variable values that satisfy all constraints
- Optimal value: Feasible solution with the best objective value

Integer (Binary) variables in optimization:
- Fixed charges in objective function
- Define Min & Max values for variables
- Constraints to choose among options
- Constraints requiring same/opposite decision
- If-then constraints

W12 Module 15 – Optimization Part 2:
Statistic Point of view
- x_{ij} are variables
- a_j are constant coefficient

Optimization point of view
- x_{ij} are constant coefficient
- a_j are variables

Optimization in linear regression:
- Variables: coefficients $a_0 \dots a_m$
- Constraints: None
- Objective Function: Minimize SSE

Other Models with same Variables & Objective function with different Constraints:
- LASSO Regression: $\sum_{i=1}^{i} |a_i| \leq t$
- Ridge Regression: $\sum_{i=1}^{i} (a_i)^2 \leq t$
- Elastic Net: $\lambda \sum_{i=1}^{i} |a_i| + (1-\lambda) \sum_{i=1}^{i} {a_i}^2 \leq t$

Optimization in Logistic regression:
- Variables: coefficients $a_0 \dots a_m$
- Constraints: None
- Objective Function: Minimize prediction error

Optimization in SVM (Hard Classification):
- Variables: coefficients $a_0 \dots a_m$
- Constraints: Each point is correctly classified $\left(\sum_{i=1}^{m} a_i x_{ij} + a_0\right) y_j \geq 1$
- Objective Function: maximize distance between vectors, thus minimize $\sum_{i=1}^{m} (a_i)^2$

Optimization in SVM (Soft Classification):
- Variables: coefficients $a_0 \dots a_m$
- Constraints: None
- Objective Function: maximize distance between vectors and minimize errors. Minimize

$\sum_{j=1}^{n} \max\{0, 1 - (\sum_{i=1}^{m} a_i x_{ij} + a_0) y_j\} + \lambda \sum_{i=1}^{m} (a_i)^2$

Optimization in Exponential smoothing model:
- Variables: $\alpha\ \beta\ \gamma$
- Constraints: $0 \leq \alpha \leq 1$ & Same for $\beta\ \gamma$
- Objective Function: Minimize SSE.

Optimization in ARIMA:
- Variables: $\mu\ \emptyset\ \theta$
- Constraints: None
- Objective Function: Minimize SSE.

Optimization in k-means clustering:
- Variables:
  - z_{jk} cluster centers coordinates
  - y_{ik} 1 if point in cluster k else 0
- Constraints:
  - $\sum y_{ik} = 1$ each point in 1 cluster
- Objective Function: Minimize total distance from data points to cluster centers

Classification of Mathematical optimization models:
Linear Program:
- Objective function is a linear function
- Constraints defined by linear equations
- Easy and fast to solve

Convex Quadratic Program:
- Objective function is a Convex Quadratic function (minimize $f(x)$ or maximize $-f(x)$)
- Constraints defined by linear equations
- Easy and fast to solve (but slower than linear programs)

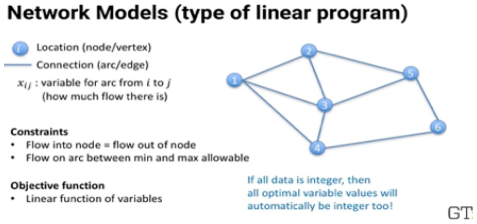
Convex Program:
- Objective function is a Convex (minimize $f(x)$) or Concave (maximize $f(x)$) function
- Constraints defined by convex equations
- Easy to solve but solution takes a long time

Integer Program:
- Objective function is a linear function
- Some (or all) variables restricted to integer values (including binary)
- Constraints defined by linear equations
- More difficult to solve

General Non-Convex Program:
- Objective function is not convex or linear function
- More difficult to solve

What if our problem is too hard?
- Heuristic
  - Rule of thumb process
  - Usually gives good solutions

Network program (Type of linear programs):



Uses of network models:

- Shortest path problem (Google maps, GPS)
- Assignment problem (Which worker gets which job to maximize efficiency
- Maximum Flow problem (how much oil can flow through complex network of pipelines

Accounting for uncertainty in optimization models:

1. Conservative modelling
   - Add a margin of error for constraints (value or probability).
2. Scenario modelling
   - Run multiple scenarios, then
     1. Run model to fit all scenarios constraints (Robust solution). Expensive solution
     2. Run model to optimize the cost of all scenarios (including gain, loss and probability of each Scenario). can lead to very complex models

Non-Mathematical optimization models:

Dynamic program:

- System is divided into states (the exact solutions and their value)
- Decisions (choice of next state)
- Beliman's equation used to determine optimal decision (next state based on current state)

Stochastic dynamic program:

- Dynamic program but decisions have probabilities of next state

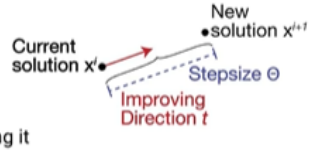
Markov dynamic program:

- Stochastic dynamic program but decisions depend only on current state

Solving an Optimization Model

Two main steps

- Create a first solution
  - Can be simple/bad/infeasible
- Repeat
  - Find an improving direction t
  - Using a step size Θ to move along it
  - New solution = old solution + Θt
- Stop when solution doesn't change much or time runs out



Convex optimization problem

- Guaranteed to find optimal solution

Non-convex optimization problem

- Not guaranteed to find optimal solution
  - Ex: converge to infeasible solution
  - Ex: converge to local optimum

Running time

- Integer programs: can be long
- Linear programs: often fast

W12 Module 16 – Advanced Models:

Non-parametric tests:

Used when the data distribution is unknown & little data is available. All we are the responses. In all tests below:

$$H_0(Null\ Hypothesis) P_A \neq P_B$$

McNemar's Test:

- Used for Yes/No Data
- Compare results of pairs of responses (where 2 different approaches used on the same thing)
- Only considers cases where A & B are different
- Checks the probability (p-value) of having such results by luck using binomial distribution.

Wilcoxon Signed Rank Test for Medians:

- Used for Numeric Data
- Assumption: distribution is continuous & symmetric
- Can also be used to check if the median of the distribution is different from m?
- Workflow:
  - Given responses $y_1 \dots y_n$
  - Rank $|y_1 - m|, \dots, |y_n - m|$ from smallest to largest
  - $W = \sum_{y_i > m} rank(y_i - m)$ = sum of all ranks where $y_i > m$
  - P-value test for W (using normal distribution)
- Can also be used for paired observations
  - Given responses $(y_1, z_1) \dots (y_n, z_n)$
  - Rank $|y_1 - z_1|, \dots, |y_n - z_n|$ from smallest to largest.

Mann-Whitney Test:

- Used for Numeric Data
- Used for 2 Datasets but not paired samples (independent data sets)
- Workflow:
  - Rank all observations together $y_1 \dots y_n$ & $z_1 \dots z_m$
  - Sum all the ranks of both datasets
  - U = Smaller of two adjusted rank sums of y & z
  - Find significance of U (Based on Normal dist.)

Matching Tests To Situations:

- Parametric tests (Mean)
  - Use exact values of data
    - Ex: Student's t-test
- Non-Parametric tests (Median)
  - Use ranks of values of data
    - Mann-Whitney, Wilcoxon tests
- Binomial-based tests (Fraction of successes)
  - Use counts of binary outcomes of data
    - McNemar's test

How many Data sets?

- One Data Set
  - Ex: Compare mean, median, etc. of one data set to a specific value m
- Two Data Set
  - Paired or unpaired
  - Ex: Compare mean, median, etc. of two data sets to each other

Bayesian Modeling

Based on conditional probability (Bayes' theorem)

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

Empirical Bayes Modeling:

- Overall distribution of something is known (or estimated broader set of observations or experts)
- Only little data is available (even a single observation)
- Make a deduction or prediction

Community Graphs:

- Used to answer questions in the analysis of large interconnected populations.
  - Disease spread through population
  - Marketing message through social media
  - Words spread between languages
- Things spread quickly in sub-populations
- Focus on automated ways of finding highly interconnected sub-populations.

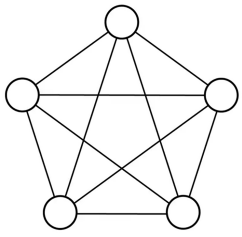
Communities

Community

- a set of circles that's highly connected within itself

Graph

- Circles = nodes/vertices
- Lines = arcs/edges
- Clique = a set of nodes that all have edges between each other



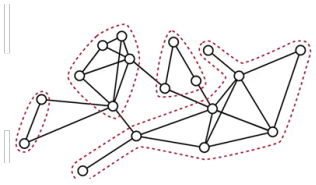
Louvain Algorithm:

- Used to decompose a graph into communities
- Heuristic Method (Fast and give good solutions, but not guaranteed to find best solution)
- Based on maximizing the modularity of a graph
- Modularity is a measure of how well the graph is divided into communities that are connected a lot internally but not connected much between each other

Maximize the modularity of a graph

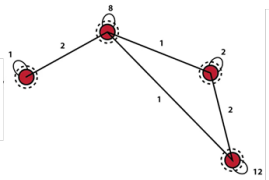
- a_{ij}: weight on the arc between nodes i and j
- w_i: total weight of arcs connected to i
- W: total weight of all the arcs
- Modularity = $\frac{1}{2W} \sum_{i,j \text{ in same community}} \left(a_{ij} - \frac{w_i w_j}{2W}\right)$

Louvain Algorithm



Step 0
Each node is its own community

Step 1
Repeat…
Make biggest modularity increase by moving a node from its community to an adjacent node's community
…until no move increases modularity



Step 2
Each community is a super-node
Repeat Step 1 using super-nodes

Neural Networks & Deep learning:

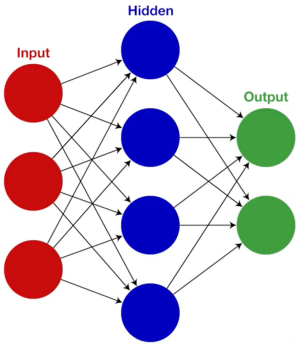
- Used to react to patterns that we do not understand
- Proven good in speech, writing & image recognition
- Neural Networks:

Neural Networks

3 levels of neurons:
- Input level
- Hidden level
- Output level

Each neuron:
- Gets inputs from previous layer
- Calculates function of weighted inputs
- Gives its output to next layer

Weights/functions updated based on correctness of results



- Example:
  - Each Input: 1 Pixel (1 piece of information)
  - Each output: 1 letter/number (result = highest result)
- Neural Networks often do not give best results:
  - Requires a lot of data to train
  - Hard to choose and tune the learning algorithm so the weights do not change too slowly or too quickly.
- Deep Learning:
  - Similar to a neural network but with many layers ("Deep")
  - In practice, seems to be a better approach
    - Relative Success in
      - Natural language processing
      - Speech recognition
      - Image recognition

Us – Against – the – data:

- Descriptive models
  - Get an understanding of reality
- Predictive models
  - Find hidden relationships
  - Predict the future
- Prescriptive models
  - Find the best thing to do
- Assumes the system does not react

Game Theory:

What if the system reacts intelligently? (We need to consider all sides of the system)

Examples

Pricing decisions	Government tax policy	Employee incentive policies	Auctions bidding	Supply chain coordination
Competitor reaction: change their price	Company reaction: how to store and spend their money	Employee reaction: change behavior	Competitor reaction: proactively adjust bid	Cooperative & competitive (negotiate payments)

- Competitive decision making (competition)
- Cooperative decision making (cooperation + competition)

Other Components:

- Timing:
  - Sometimes decisions are made simultaneously and can't be changed once made (biding). Need to consider Strategy & Counter-Strategy, etc.
  - Sequential Game: decisions are made sequentially
- Types of strategies:
  - Pure strategy: Just one choice all the time
  - Mixed strategy: randomize decisions according to strategy (rock-paper-scissors)
- Information levels:
  - Perfect information: know all about everyone else's situation (Playing chess)
  - Imperfect information: some have more information than others – not symmetric (gas station)
- Zero Sum & Non-Zero Sum Games:
  - Zero-sum: whatever one side gets, the other loses (ex. Rock-paper-scissors)
  - Non-Zero Sum: total benefit might be higher or lower. (ex. Economics)

Determine the best strategy by **optimization models**.

Natural Language Processing (NLP):

- Speech-to-text
- Recognizing named entities and co-references
- Determining meaning in context
- Understanding sentiment
- Inverse problem: choose words to convey message

Survival Models (Insurance Industry):

- Predict the probability of an event happening (or not)before a certain time
  - Based on predictor variables

Cox proportional hazard model

- Given (for each data point)
  - Predictor variables $x_1,\ldots,x_n$
  - Event time y

$e^{\beta_j x_j}$ = probability multiplier due to factor j

$$h(t) = h_0(t)\, e^{(\beta_1 x_1 + \cdots + \beta_n x_n)} = h_0(t) e^{\beta_1 x_1} e^{\beta_2 x_2} \ldots e^{\beta_n x_n}$$

β_j: coefficient for each predictor variable

Baseline (if all predictor variables are zero) probability of event happening at time t

Probability of event happening at time t given specific values of predictors (and coefficients)

$\beta_j = 0$: x_j has no effect
$\beta_j > 0$: positive x_j, higher probability
$\beta_j < 0$: positive x_j, lower probability

Censored Data:

- No data before or after a specific time
- No data after a certain number of observed occurrences

Gradient Boosting:

- Using additional models to augment model
- Start with a starting model
- Other models – Fit using gradient information
- Used with factor-based models

Basic Idea