1 Combinatorics

- The study of the number of ways we can arrange a set of elements
- Permutations - order matters
  - ${}_nP_r = \frac{n!}{(n-r)!}$
  - i.e. count the number of ways runners could split medals
- Combinations - order does not matter
  - ${}_nC_r = \binom{n}{r} = \frac{n}{r!(n-r)!}$
  - Symmetrical: ${}_nC_r =_n C_{n-r}$

2 Bayesian Inference

- Every set has a set of outcomes and at least two subsets (itself, null)
- $A \cup B = A + B - A \cap B$
- $P(A|B) = \frac{P(A \cap B)}{P(B)}$
- $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

3 Probability Distribution

- Possible outcomes an event can take and the frequency of occurrence
- P(X = x). X = actual outcome of an event and x = one of the possible outcomes.
- $\sigma^2 = E((X-\mu)^2) = E(X^2) - \mu^2$
- Discrete - finite outcomes
  - Uniform - each outcome has an equal chance
  - Bernoulli - event with two outcomes
    - $E(X) = p$
    - $Var(X) = p(1-p)$
    - $P(X) = p^x(1-p)^{1-x}$
  - Binomial - multiple bernoulli trials
    - $E(X) = np$
    - $Var(X) = np(1-p)$
    - $P(X) = \binom{n}{x}p^x(1-p)^{n-x}$
  - Poisson - frequency at which an event occurs
    - $E(X) = Var(X) = \lambda$
    - $P(X) = \frac{\lambda^x e^{-\lambda}}{x!}$
- Continuous - infinite outcomes
  - Normal: $E(X) = \mu, Var(X) = \sigma^2$
  - T - small approximation to normal distribution, fat tails
  - Chi-squared - for goodness of fit
  - Exponential - events rapidly changing early on
    - $E(X) = \frac{1}{\lambda}$
    - $Var(X) = \frac{1}{\lambda^2}$
    - $P(X) = \lambda e^{-\lambda x}$

4 Statistics

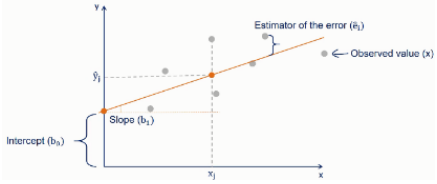
- Population: collection of all items of interest (N)
- Sample - subset of a population (n)
- Sample must be random and representative of population
- Descriptive
  - Two types of data (categorical, numerical)
  - Two types of measurement - Qualitative: nominal and ordinal - Quantitative: interval and ratio
  - Pareto Principle - 80% of the effect comes from 20% of the causes
  - Mean > Median: Skewed right (+)
  - Mean < Median: Skewed left (-)
  - Mean = Median: Not Skewed
  - Mode: peak of distribution
  - Coefficient of variation: $\frac{\sigma}{\mu}$, useful when comparing 2+ datasets
  - $Cov(X,Y) = \Sigma_{i=1}^{n} \frac{(x_i - \overline{x})(y_i - \overline{y})}{n-1}$
  - $Corr(X,Y) = r = \frac{Cov(X,Y)}{\sigma_x \sigma_y}$ $-1 \leq r \leq 1$
  - Correlation does not imply causation
- Inferential
  - Using probability theory to predict population values using sample data
  - Central Limit Theorem - if you take large random samples from a population, then the distribution of the sample means will be approximately normal regardless of the distribution of the population. Hence, sampling distribution $\sim N(\mu, \frac{\sigma^2}{n})$
  - Standard Error: standard deviation of sample mean distribution
  - Good estimators have two properties:
    - Efficiency: smallest variance
    - Bias: expected value = population parameter
  - Confidence Intervals
    - how confident are you population parameter is contained in interval surrounding sample estimate
    - if population variance is unknown, use the t-dist: $\overline{x} \pm t_{n-1,\alpha/2}\frac{s}{\sqrt{n}}$
    - if two means, independent samples, and variance is known: $\sigma^2_{diff} = \frac{\sigma_e^2}{n_e} + \frac{\sigma_m^2}{n_m}$ $(\overline{x} - \overline{y}) \pm z_{\alpha/2}\sigma_{diff}$
    - if two means, independent samples, and variance is unknown: $s_p^2 = \frac{(n_x-1)s_x^2 + (n_y-1)s_y^2}{n_x n_y - 2}$ $(\overline{x} - \overline{y}) \pm t_{n_x+n_y-2,\alpha/2}\sqrt{\frac{s_p^2}{n_x} + \frac{s_p^2}{n_y}}$
  - Hypothesis Testing
    - Hypothesis: idea that can be tested
    - Research is trying to reject the null Hypothesis
    - Type I error: rejecting a true null hypothesis (α)
    - Type II error: accepting a false null hypothesis (β)
    - p-value: smallest level of significance we can still reject the null hypothesis given the observed sample statistic.

5 K-Nearest Neighbors

- For every test observation find its k closest training observations. Then, take a majority vote or average on their class labels and assign its label to the test observation.
- $P(Y = j) = \frac{1}{K}\Sigma_{i \in N_0} I(y_i = j)$

6 Simple Linear Regression

- A linear approximation of a causal relationship between two variables
- $y = \beta_0 + \beta_1 x_1 + \epsilon$
- Correlation is the degree of relationship between two variables whereas regression is how one variable affects another
- Correlation is symmetric. Regression is one-way
- Correlation is visualized as a single point on a scatter plot. Regression is a best fit line on a scatter plot.



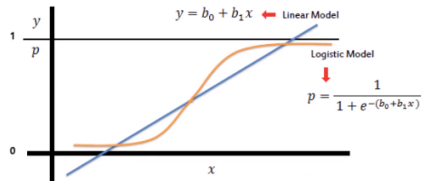
- $SST = \Sigma_{i=1}^{n}(y_i - \overline{y})^2$: total variability of dataset
- $SSR = \Sigma_{i=1}^{n}(\hat{y}_i - \overline{y})^2$: variability explained by regression
- $SSE = \Sigma_{i=1}^{n}(\hat{y}_i - y_i)^2$: unexplained variability
- $SST = SSR + SSE$
- Ordinary Least Squares (OLS) to estimate parameters such that SSE is minimized.
- $R^2 = \frac{SSR}{SST}$: Proportion of total variability explained by regression
- What's a good R^2?:
  - goodness of fit: [0.2,0.9]
  - physics/chemistry: [0.7, 0.9]
  - social sciences: [0.2]
  - generally depends on the topic and number of independent variables

7 Multiple Linear Regression

- $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \epsilon$
- more variables, more explanatory power
- $R^2_{adj} = 1 - \frac{SSE/(n-K)}{SST/(n-1)}$
- F-Test for overall significance:
  $H_0 : \beta_1 = \beta_2 = ... = \beta_k = 0$
  H_1: at least one $\beta_i \neq 0$
- OLS Assumption
  - Linearity: relationship between the dependent variable y and independent variable x is linear.
  - Normality: All variables must be multivariate normal otherwise you need a non-linear transformation.
  - No multicollinearity: All variables are independent of each other
  - No autorrelation: Observations must be independent of each other
  - Homoscedasticity: Variance of the error terms given x is constant
- number of dummy variables = number of categories - 1. Each dummy has a value of 0 or 1.
- Feature Scaling (Standardization)
  - to ensure no one variable is more important than the other
  - Ex. Euro Exchange rate vs. daily trading volumes
- Underfitting and Overfitting
  - Overfitting - training accuracy is high but testing accuracy is low. Model is too foucsed on the training set that it has "missed the point". You modeled the noise.
  - Underfitting - Model has not captured the undelrying logic of the data. To overcome, split the data into training, testing, and shuffle the data.
  - Bias - Variance Tradeoff: A balance between an underfitted and an overfitted model broken by model complexity.

8 Logistic Regression

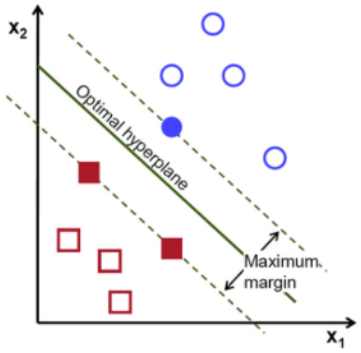
- Regressing probability of a categorical variable
- Assumptions same as SLR except linearity is violated
- Can't use linear regression because output would be outside [0,1]
- So use sigmoid: $\mathbb{R} \rightarrow [0, 1]$
- PDF Derivation
  - Take sigmoid: $\delta(t) = \frac{1}{1+e^{-t}}$
  - Take SLR: $t = \beta_0 + \beta_1 x$
  - Plug in: $\delta(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$
  - Logit: $log(\frac{p(x)}{1-p(x)}) = \beta_0 + \beta_1 x$



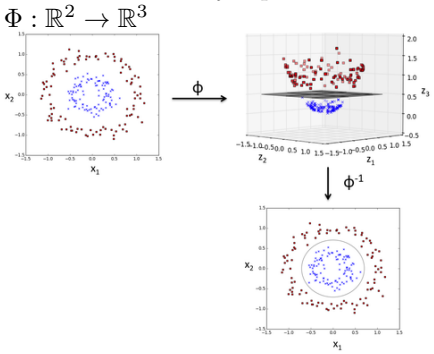
- Use MLE for parameter estimation
  - Estimates how likely model describes real underlying relationship of variables
  - Bigger the likelihood, higher the probability the model is correct
  - Easier to maximize log likelihood
- Log likelihood ratio test: for overall model significance
- Confusion matrix: how confused model is

9 Support Vector Machines

- SVM attempts to find a hyperplane that separates classes by maximizing the margin
- See diagram. The filled in points are the support vectors of the decision hyperplane.
- So, really low x1 and x2 would be classified as a square and realy high x1 and x2 would be classified as a circle.

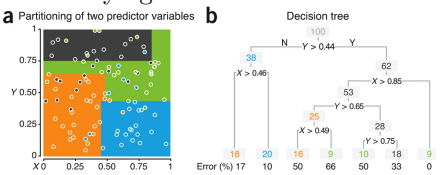


- If we have a non-linear decision boundary, we'll use the kernel trick: map linear non-separable inputs into a higher dimension where they become more easily seperable.
- $\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$



10 Decision Trees

- Partition the feature space into regions and computing the mean/mode of the training responses (regression) in that region in classifying the test observation.



- Algorithm
  - High Level: Split target class into the purest possible children nodes. Measure of purity is called the information.
  - Node purity: node contains predominantly observations from a single class
  1. Use recursive binary splitting to grow a large tree. Stop when each terminal node has fewer than some minimum number of observations.
  2. Node impurity splitting metrics:
     - Let $\hat{p}_{mk}$ be the proportion of training observations from the m^{th} region from the k^{th} class. We want small values for gini index, cross entropy, error.
     - Categorical Target
       - Information Gain: Entropy before - Entropy after
       $Entropy = -\Sigma_{k=1}^{K} \hat{p}_{mk} log \hat{p}_{mk}$
       - Gini Index:
       $G = \Sigma_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$
       - Classification Error:
       $E = 1 - max(\hat{p}_{mk})$
     - Continuous Target
       - Variance Reduction:
       $S(T) - S(T, X) =$
       $\frac{1}{n}\Sigma_{i=1}^{n}(y_i - \overline{y})^2 -$
       $\Sigma_{c \in X} P(c) S(c)$
  3. Prune the tree or use Random Forest to prevent overfitting

11 Random Forest

- To increase the predictive power of decision trees.
- Algorithm
  1. Construct B regresion trees using B bootstrapped training sets (sampling with replacement), and average the resulting predictions or take a majority vote for classificaiton problems
  2. Let trees be deep and unpruned
  3. For each tree, use 2/3 as training, 1/3 as OOB observations
  4. Predict response on the OOB

observations. Repeat on all B trees
5. Average predicted responses or take majority vote to get a single OOB prediction for each observation
6. Compute the overall OOB MSE or classification error

- Random Forest de-correlates trees:
  - Each time a split is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors ($m \approx \sqrt{p}$)
  - Trees will not look similar to each other since strong predictors, which otherwise would be present in most trees, will be removed
- Can find important variables
  - Record the amount the RSS/Gini index has decreased due to splits over a predictor, averaged across all B trees.
  - Large values indicate important variables

12 Cluster Analysis

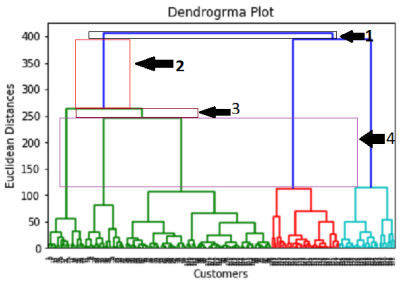
- Dividing observations into groups based on features
- Goal is to maximize similarity of observations within a cluster and maximize dissimilarity between clusters
- Curse of dimensionality: an observation has no nearby neighbors (i.e. p > n). This will drastically increase the error rate and computation time. Use Manhattan distance if this occurs.
- Distance Metrics:
  - Euclidean
    - Find shortest path between points
    - $\sqrt{\Sigma_{i=1}^{k}(x_i - y_i)^2}$
  - Manhattan
    - Find shortest zig-zag path between points
    - $\Sigma_{i=1}^{k}|x_i - y_i|$
  - Hamming
    - Find distance between two binary data strings
    - Used for categorical variables
    - $D_H = \Sigma_{i=1}^{k}|x_i - y_i|$
- Examples:
  - Market segmentation
  - Data Exploration
  - Image segmentation
  - Object recognition

13 K-Means Clustering

- Algorithm
  1. Choose number of clusters
     - WCSS: make as small as possible
     - Elbow method: plot of WCSS vs. num clusters. Look for diminishing improvements
  2. Specify the cluster seeds
  3. Assign each point to a centroid based on distance
  4. Adjust centroids
  5. Repeat steps 4 and 5 until a stopping criterion is met. It can be (1) centroids don't change much, (2) points remain in same cluster, (3) max number of iterations is reached
- Pros
  - Simple to understand
  - Fast to cluster
  - Easy to implement
  - Always yields a result
- Cons
  - Need to pick a K
  - Sensitive to initialization → k-means++
  - Sensitive to outliers → remove them
  - Produces spherical vs. elliptic solutions
  - Must standardize to put variables on equal footing

14 Hierarchical Clustering

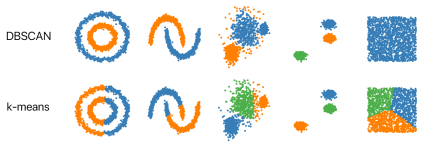
- Two forms of Clustering
  1. Agglomerative (bottom-up)
     - Easy to solve
     - Start at bottom and pair closest observations into a cluster. Use euclidean distance to iteratively group clusters until there's only one.
  2. Divisive (top-down)
     - Have to consider all possibilities until there's one cluster
- Dendrogram
  - Tree representation. Start from bottom and work your way to the top. Ite tells you how similar clusters are to each other based on the distance between the links.
  - Draw the line when the distance between the clusters is too big



- Pros
  - Shows linkages between clusters
  - Understand the data much, much better
  - No need to preset the number of clusters (k-means)
- Cons
  - Huge dendrogram
  - Computationally intensive

15 DBSCAN

- Density-based spatial clustering of applications with noise
- Clusters are continuous regions of high density; low density regions separate clusters.

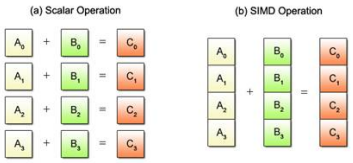


- Two hyperaparameters to specify
  - Epsilon: distance metric to locate points/check density in neighborhood of any point Optimal: plot distance of every observation and neighbors. Select epsilon at point of maxmimum curvature.
  - minPoints: minimum number of points clustered together for a region to be considered dense
- Pros
  - Outliers easily identifiable
  - Can take irregular shapes
  - Don't need to preset number of clusters
- Cons
  - Bad for sparse datasets
  - Sensitive to hyperaparameters
  - Can't partition for multiprocessing

16 Linear Algebra

- Everything is a tensor
- Tensors have ranks (number of axes)
- Scalar: 1x1. Rank=0
- Vector: mx1. Rank=1
- Matrix: m*n. Rank=2
- Triad: m*n*k. Rank=3
- Why is linear algebra useful?

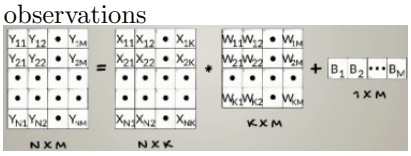
- Vectorization for computational efficiency
  - Can by-pass using a for-loop and take advantage of the SIMD paradigm.
  - SIMD: Single Instruction - Multiple Data, a method for combining multiple operations into a single computer instruction



- Dimensionality reduction
  - "curse of dimensionality": when p > n. The sample is too small and too many inputs can dramatically impact model performance.
  - As a solution, use matrix factorization (LU, QR, Eigendecomposition, SVD) and PCA
- Computer vision
  - All images are stored as a matrix with values between 0 and 255. Colored ones stored on the RGB system have three layers, stored as a tensor of rank 3: m*n*3.

17 Deep Learning

- Types of Machine Learning
  - Supervised
  - Unsupervised
  - Reinforcement
- Linear Model
  - $Y = XW + B$
  - Multiple inputs, outputs, observations



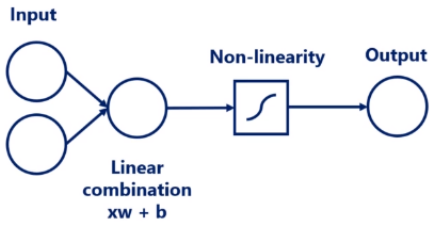
  - Deep Neural Networks are helpful for finding nonlinearly separable boundaries to data

18 TensorFlow

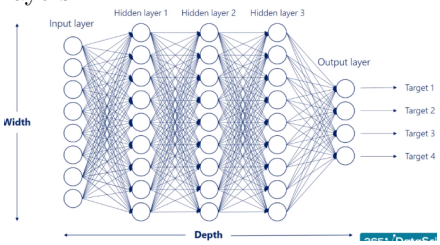
- Software developed by Google that utilizes GPU in addition to CPU for deep learning models. Optionally, it can use TPUs for models.
- Why use GPUs over CPUs?
  - Are optimized for parallel computing
  - Have thousands of cores
  - Has multiple hyperthreads per core
- Why use CPUs over GPUS?
  - CPUs process data sequentially
  - They do not know what instruction will be next (i.e. input from keyboard, mouse, ...)
  - Has resources to manage an Operating System

19 Neural Networks

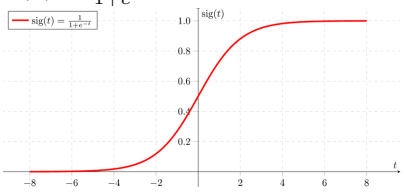
- Neural Network: set of algorithms (supervised, unsupervised, reinforcement) trying to recognize patterns and relationships within data



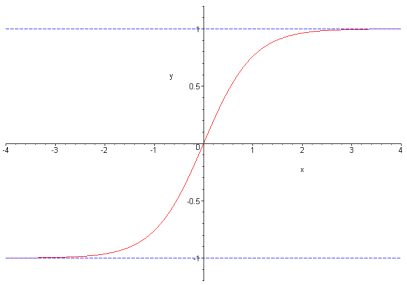
- Deep Neural Network: A neural nework with one or more hidden layers



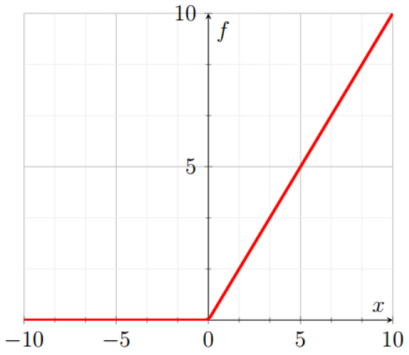
- Hyperparameters: pre-set by the practitioner
- Parameters: set by optimization
- Why do we need non-linearites?
  - Two consecutive linear transformations are equal to a single one, meaning hidden layers are useless
  - Use non-linearities to find complex relationships
- Four common activation functions. All are monotonic, continuous, and differentiable
  - Sigmoid: $(-\infty, +\infty) \rightarrow (0, 1)$
    $f(x) = \frac{1}{1+e^{-x}}$



  - Tanh: $(-\infty, +\infty) \rightarrow (-1, 1)$
    $f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$

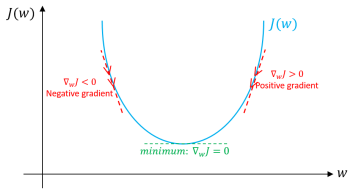


  - ReLu: $(-\infty, +\infty) \rightarrow (0, +\infty)$
    $f(x) = max(0, x)$



  - Softmax: $(-\infty, +\infty) \rightarrow (0, 1)$
    $f(x) = \frac{e^{z_i}}{\Sigma_{j=1}^{K} e^{z_j}}$
    Softmax is most often used in the final output layer. The output layer has values between 0 and 1. The function transforms a bunch of numbers into a valid probability distribution.
- Backpropagation
  - Technique to update the weights and biases in a neural network in a way to minimize the loss function
  - Vanishing Gradient: gradient diminishes as it propagates backwkard through the network. By the time it reaches layers close to the input, it may have little effect. Use ReLu as a solution because the derivative will not be anything near 0.
  - Exploding gradient: gradient exponentially increases as it's backpropagated through the network. Solution (1) gradient clipping if their norm exceeds a threshold, (2) redesign the network and use smaller batch sizes and LSTMs
- Loss function - how well model's outputs match the true outputs
  - Regression: L2 RSS Norm - $L = \Sigma_{i=1}^{N}(y_i - \hat{y}_i)^2$
  - Classification - Cross Entropy - $L = -\Sigma_{i=1}^{N}\Sigma_{j=1}^{M} y_{i,j} ln(p_{i,j})$
- Gradient Descent: algorithm to find the minimum of a loss function. Takes small steps in

direction of steepest descent.



- $x_{i+1} = x_i - \gamma f^{`}(x_i)$
- γ is the learning rate (speed of minimization)
- If learning rate is too low, it wil take forever to converge. If too high, we'll oscillate around the minimum but not converge.

- Batch size: number of samples needed to update parameters
  - (BGD) Batch gradient descent: use total number of training examples
  - (SGD) Stochastic gradient descent: batch size = 1 training example
  - (MGD) Mini-batch gradient descent: batch size = subset of total number of training examples.
  - MGD > SGD > BGD. BGD may not fit into memory and get stuck at a local minima, SGD with fewer data points jerks model out of local minima but is very noisy, MGD is a balance between the two.
- Epoch: one full pass through training dataset

20 Model Evaluation and Selection

- Bias-Variance Tradeoff
  - $y = f(x) + \epsilon$
  - ϵ = Bias + Variance + Irreducible error
  - Bias: How often model's predicted values come to the true underlying f(x) values
  - Variance: how often does prediction error change as you change the training dataset
  - Irreducible error: due to inherently noisy observation process
- Model Complexity and Overfitting
  - Overfitting remedies:
    - Regularization (Shrinkage): Add a penalty to the objective function. The penalty shrinks coefficients.
    - L1 (Lasso): $\Sigma_{i=1}^{N}(y_i - \hat{y}_i)^2 + \gamma\Sigma_{j=1}^{p}|\beta_j|$
    - L2 (Ridge): $\Sigma_{i=1}^{N}(y_i - \hat{y}_i)^2 + \gamma\Sigma_{j=1}^{p}\beta_j^2$
    - L1 can zero-out parameters, L2 can shrink but not zero-out
- Interpretability
  - In the real world, explaining the model is important and choosing a simpler model in fields like healthcare, IRS, ... is important.