* Induction
  + Generalize from a specific example to tell the future if you will
  + \*\*Carful because the past doesn’t always predict the future
  + \*\*Bias – any bases to choose among generalizations other than the data itself. Help us choose beyond just the data, comes from outside the data.
    - It is essential to learn
    - In the absence of bias you cannot learn.
* Deduction
  + Given a general principle tell the truth
* Unsupervised Learning
  + No external forces to help learning
* Supervised
  + Some external direction or help
* Bias – Variance Trade Off
  + Strong – number of generalizations decrease, decrease variance
  + Weak – number of generalizations increase, increase variance
* \* As humans we adapt and change our bias constantly. It is what allows us to learn.
* How do you know what you’re doing is any good…when you make a guess to the data?
  + Bias
    - Any basis for choosing one decision over another, other than strict consistency with past observations
* If you have no bias you cannot go beyond mere memorization
  + The power of a generalization system follows directly from its biases
  + w/o bias you can only memorize
* Unbiased Language -- has no bias – it can represent everything… any set can be given a name.
* Unbiased Generalization Procedure – it can pick any set…
* Version Space Algorithm
  + G – everyone
  + S – everyone we know to be evil
  + Everything else
* If you have an Unbias language and Unbias procedure – you cannot learn, you can only memorize.
* Sources of Bias in Learning
  + The representaitoin language cannot express all possible classes of observation
  + The generalization procedure
* Overfit – become so good at classifying test data you loose ability to correctly classify new data
  + Avoid
    - Stop early – either number or proportion reaches a limit
  + Prune it – POST Pruning
    - Hold some data back – then run it back through and see how the held data works
    - Cut a branch and see if it gets better… If it does cut the branch
* Questions:
  + 25-40
  + 50-300
  + 2-5
  + 5-30
  + 500-900
  + 1000-2000
  + 1600-1700
  + 60-120
  + 10000-20000
  + 150000-900000
* Why didn’t you guess –inf to inf?
  + Because we’re bias
  + And we hold onto our biases
    - Don’t want to look stupid
    - I’m close
    - I know the answers
* Information gain == difference in entropy
* == entropy before – entorypy after
* Why Entropy over Accuracy?
  + Entropy we just need to change the distribution
  + However if using accuracy…we are trying to change the majority with the split
    - This could be very hard to do

Eager Learner – once model is created you can discard the data.

Lazy Learner – can’t discard the data

Batch Learner / Inline or Incremental Learner

* \* know the biases of each algorithm
* K-Nearest Neighbor:
  + Bias: things that look like me should be classified like me.
  + Assumption : we can put things into some meaningful space and the distance around them is meaningful.
  + Assumption : we have continuous space
  + ….What is the distance between red and green?
    - How do you measure these things.
    - Could do ordinal – discrete but the difference makes sense.
* DTree – bias : build small trees, no data, set up model, re run to change
* K – bias: , incremental, lazy, needs data
* NB – Inbetween K, and Dtree
  + Uses model
  + But can update model easily
  + Doen’t have to keep the data
  + P(h|D) – find H that maximizes this thing
* Bayes
  + Way different, from the others. Its run on probabilities
  + Prier is a big deal…tells you how it changes as data becomes available. Probabilities drive the decisions you take.
    - List assumptions:
      * Congunction of att values
      * Att are independent given the class
        + NB is still robust
      * Est probabilities with frequencys
* -Reminders
  + Tension between the prior and the data
  + Handles new data easily
  + Incremental
* M-estimates
  + Get ride of the zero probability thing… Like you will never play tennis when its rainy
* K- nearest
  + Bias – what looks like me should be classified like me
    - The distance metric
  + Incremental
  + Harder to explain why now vs Decision Tree
* Black Box stuff
* Sub-simbolic
* Perceptron
  + Must be linearly separable
    - Otherwise they cannot converge
* Take away
  + Every time you touch the data can introduce bias
  + If things are too good to be true they are
  + Simsons paradox
    - Confounding variable
    - Cause things to look differently
    - Like seasonality
* Data Manipulation
  + Why?
    - Unsable
    - Too many attibutes
    - Screwed data
    - Missing data
  + How
    - Type converstion
    - Discretization
      * Equal height
      * Equal increment
  + Good data preparation is key to predicting valid and reliable mdoels
    - The key is it takes a lot of work to make sure the algorithms work
* F – give us a measure of Prescision and recall
  + 1 is the best
  + lower is less good
  + tree
  + bayes
* visulation
* understanding
  + manip
  + pca
* measures
* quote and reacte
* Combine Learning Algorithms
  + Why
    - A number of novices is better than one expert.
    - Help overcome individual biases
* Ensembling
  + Because we have difference learners looking at the data…we are trying to see if we can improve the result of what a single learner would do.
* Bagging – Reduce variance cause by the data
  + Idea to reduce the variance on a learning algorithm
  + How and why reduce variance
    - If there is a lot of variance in the LM it get averaged out over the set of learning algorithms. The hope is that each one will learn something of the structure of the data and then together they can speak intelligently about the data.
  + There are things where bagging helps and where it doesn’t’
  + But if the LM has little variance then this will not help
* Boosting – Improve Accuracy
  + Can over fit the data – be carful
  + There is a fumula to calculate the change in weight
  + Weighting in vote is 1/log(beta)
    - Error = sum y != f(x)Mn
    - Beta = error / 1-error
  + Give more weight to the data that was incorrect
    - Get some more of the hard ones right
* Stacking
  + Data->several algorithms->get models->run models get new predictions -> use preditions as a new data set -> run new data through a new algorithm
  + Hope we get some regularity
* Ensemble, boosting, stacking 🡪 improve accuracy
* Delegating
  + Like boosting but different
* Arbitrating
* NFL
  + When you take an arbitrary learner against all learning tasks the Generalization task will be no better than guessing…it will be no better as guessing. If you do good over here you will must do poorly somewhere else.
  + Across all learning task the generalization of a learner across all learning taks is no better than random
  + This has helped us make good statements when we are making comparison statements.
* NN
  + Stopping Criteria
    - Ideal: e == 0
    - SUM: (Ti – Oi) ^2
    - E < e
    - dE < e – Change in error drops below some threshold
    - #iterations < N
  + Might want to normalize our data
* How do we classify data with a label? How do we group data in some way that makes some sort of sense.
* Clustering
  + Unsupervised
  + How
    - Paritional – has distance metric
    - Hierarchical – has distance metric
    - Symbolic – based on probablilites
* K-Means
  + Single partition
  + K – number fo paritions
  + Numeric data only
  + K random places in space to start
  + Points act as gravity wells
    - Given the points recomputed a center point
    - Rinse and repeat
  + Issues
    - Results depend on starting states
    - No really metric for determining which state is better
    - Outliers
    - Does not scale very well
      * Computing a lot of distances
    - Local Minimum – everything is equal distance
    - You have to pick the number of clusters
    - All items forced into a single cluster
  + Adv
    - Simple
    - Items automatically assigned to cluster
* K-Medoids
  + Follow and use data points
  + Know have a quality measure
* **Hierartical Methods to Cluster**
* 1) everyone in own cluster, merg the closest two
* 2) everyone in the same cluster, figure out a good way to split
* WE’ll look at 1.
* Min Link, Average Link, Max Link – How we compute the distance between cluster with multiple points.
* When to stop…Dendrogram
  + Right before big jumps
  + Pick a k

One bias about k-means, k-metroid

* + You have to pick K
* HClust
  + You still have to pick where to make a cut

COBWEB

* + Maximize similarities, max difference among different clusters
* Aprori
  + Define Interesting
    - Support >= min support
    - Confidence >= min confidence
  + Expensive
    - However can prune early if some item or set of items is below threshold
* Web Mining
  + Sourses of data
    - Usage
      * Who, when, where, why
    - Structure
    - Content
  + What can we look at
    - Content
      * Extract snippets from a web doc that represents the web doc
    - Strutures
      * Id interesting graph patterns or preprocessing the whole web graph to come up with metrics such as PageRank
    - Usage
      * User id, seesion creation, robot detection
* Web Usage minning
  + Discovery meaninguful patterns form data generated by client-server transactions on one or more web localities
  + Issues
    - Session id
    - Cgi data
    - Caching
    - Dynamic pages
    - Robot detection and filtering
  + Session id solutions
    - Cookies
    - User login
    - Embedded sessionID
    - IP+Agent
    - Client-side tracking
* Moveing the space to
  + The log of the odds
  + In new space
    - We say logit(p) = linear combingation
* No Free Lunch
  + Theoretical result in which every set of outcomes is equaly likely…
* Now we have a large number of algorithms but no real way to determine which task is best for which task…
  + Can we think up a way to offer support on how to choose a good algorithm, on how to use machine learning.
    - Nature of the Data
      * Not all outcomes are equal likely
    - Outcome we’re looking for
    - Meta data (num attributes, what they are looking for, missing values, extra)
      * Then run aglrothims and rate them on performance
      * Learn which are best
      * Then we can look at the structure of the data and draw conclusions on which would be best.
* Summarize:
  + Nature of the Data
  + NFL
    - Not practical
    - We don’t care against all tasks
  + Metadata
    - N-Tasks 🡪 Extract M Attributes
    - Run the Nth task against all learning algorithms and attach the best algorithm to it
    - What are the issues here
      * We need a lot of learning tasks
      * We need to pick the best learner for the meta learner problem
      * Do the attributes actually help predict the class
        + Are they meaningful across all learning tasks
      * Just gathering the meta data to run agains a meta learner might be more expensive than just trying all possible algorithms.
  + Model Characteristics
  + Landmarkers
    - Small cut down versions of the algorithm to tell how the full one will do.
* REVIEW:
* Clustering
  + Interest ways to cut
* Assison rule minning
  + Support, confidence
  + What It means if you have larg rules
* Lin/ regression
  + Changing space
* Supor vetor macines
* What model works when…
* Study Guide
* Approri – run it
* Rules large
* Not large
* Support, confidence
* Regression, how it works
* t/f - question
* Pick the right approach to solve a set of case studies
* Web minning
* Clustering, kmean, hiarticl clustering – run it, issues are
* Svm, how, why, assumptions