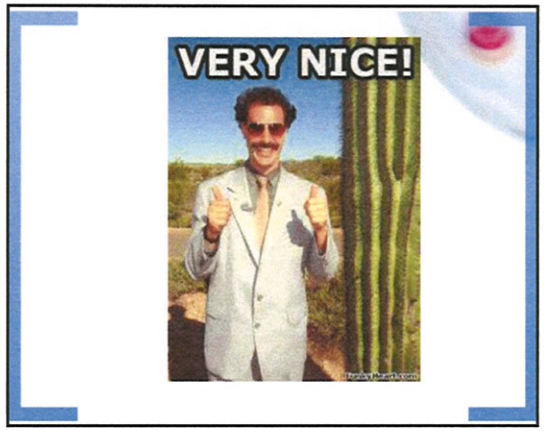
The Challenge of Big Data

Jennifer Clarke

Director, Computational Sciences Initiative



Computationa l Sciences Initiative (CSI)

A university-wide program of excellence for developing expertise and resources in Big Data and data science.

Who are we?

Jennifer Clarke, Associate Professor, Statistics, Food Science and Technology

Lisa Lightner, Administrative Wizard

Dmitri Fomenko, PhD, Staff Scientist [Redox biochem]

Sanjay Babu, PhD, Postdoctoral Researcher [metagenomics]

Big Data Consortium

Computational Sciences Initiative Advisory Board

What do we do?

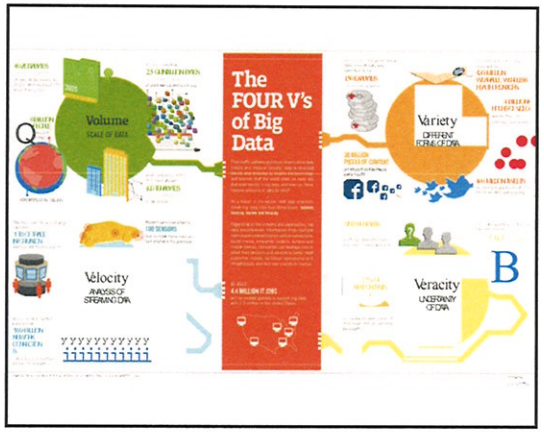
Enable interdisciplinary and basic research in data science

advocate for/develop data resources and expertise

serve as Big Data liaison between UNL & industry partners

develop both graduate and undergraduate curricula in data science

The ways in which we analyze, process, store, and interact with data are rapidly changing



The challenge facing educators and researchers is how to get from data to knowledge when data is large, noisy, and complex.

Data to Knowledge: Research

- Researchers across disciplines are discovering that it is relatively easy to collect Large amounts of data and relatively difficult to find answers to interesting questions
- Finding answers will involve collaborations with the information sciences (e.g., math, stat, ICE, CSE, bioinformatics)

Data to Knowledge: Education

- Higher education has been designed to train field-specific specialists, and Universities reward faculty with extremely specialized knowledge
- Need for cross-disciplinary knowledge and skills
- Undergraduate eduction in general and the Liberal arts in particular are under increasing pressure to demonstrate relevance (ROI)
- A coherent curriculum in data sciences and a broad emphasis in basic information skills

Foundations

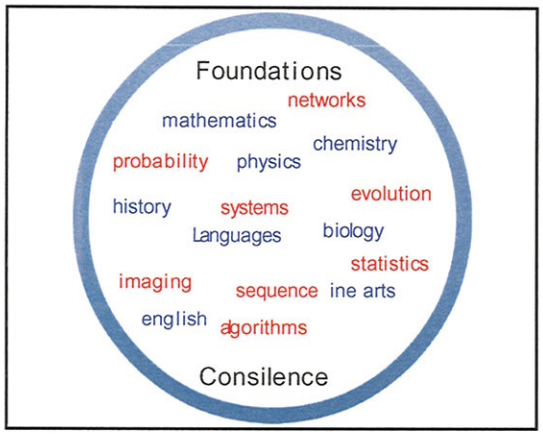
mathematics chemistry

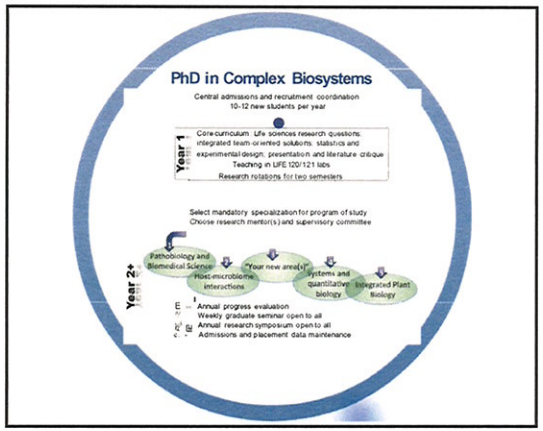
physics

history biology

languages

english **fine arts**


Foundations
networks
mathematics
chemistry
probability
physics
history
evolution
systems
Languages
biology
statistics
imaging
sequence
ine arts
english
algorithms
Consilence


PhD in Complex Biosystems
Central admissions and recruitment coordination
10-12 new students per year
Year 1
Core curriculum: Life sciences research questions; integrated team-oriented solutions; statistics and experimental design; presentation and literature critique
Teaching in LIFE 120/121 labs
Research rotations for two semesters
Select mandatory specialization for program of study
Choose research mentor(s) and supervisory committee
Year 2+
Pathobiology and Biomedical Science
Host-microbiome interactions
Your new area(s)*
Systems and quantitative biology
Integrated Plant Biology
Annual progress evaluation
Weekly graduate seminar open to all
Annual research symposium open to all
Admissions and placement data maintenance

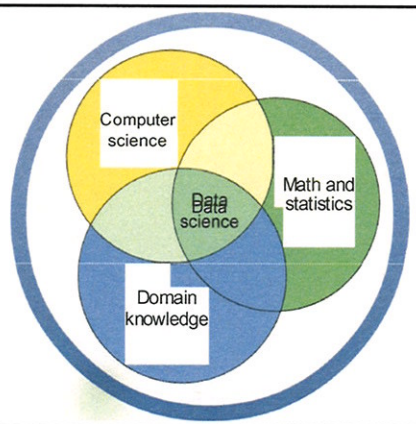
LET'S SOLVE THIS PROBLEM BY USING THE BIG DATA NONE OF US HAVE THE SLIGHTEST IDEA WHAT TO DO WITH
©marketoonist.com

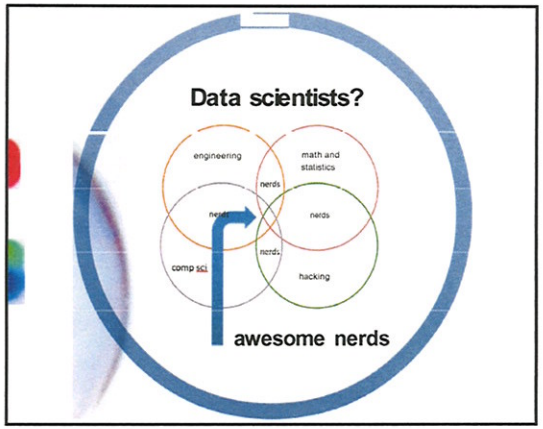
Why Big Data?

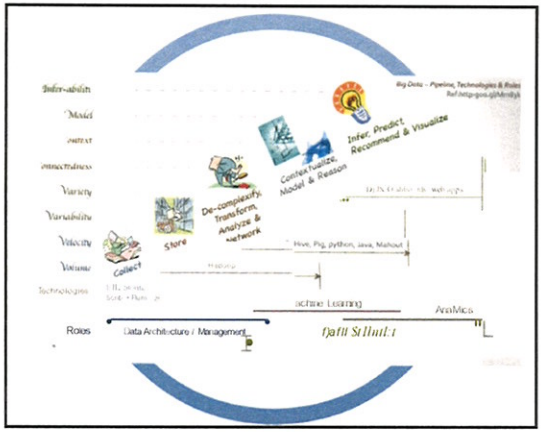
'The forward edge of science, whether it drives a business or marketing decision, provides an insight into Renaissance painting, or Leads to a medical breakthrough, is increasingly being driven by quantities of information that humans can understand only with the help of math and machines'

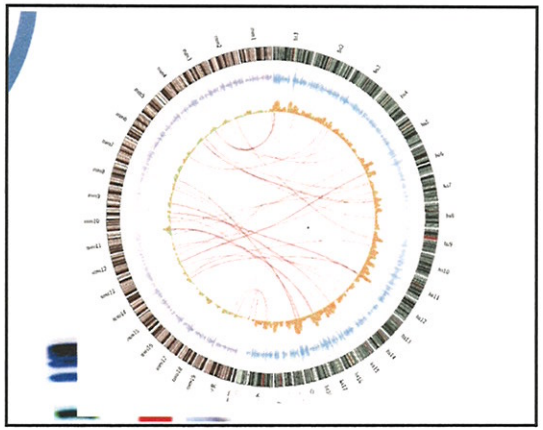
Gary King

The big data revolution is that now we can do something with the data






Infer, Predict, Recommend & Visualize
Contextualize, Model & Reason
De-complexify, Transform, Analyze & Network
Store
Collect
Hive, Pig, Python, Java, Mahout
achine Learning
AnaMics
Roles
Data Architecture / Management



Model Ensembles

Jennifer Clarke, Ph.D.

Associate Professor
Department of Statistics
Department of Food Science and Technology
University of Nebraska Lincoln

ASA Snake River Chapter, Meridian, ID, May 29 2015

Outline

What is an 'Ensemble'?

Bayes model average (BMA)

Bagging

Stacking

Boosting

What is an 'Ensemble'?

- The key idea is to take a bunch of models that may or may not represent a coherent theory and average the predictions from them to get a better predictor
- Basic observation under squared error loss:
  $E(Y - E(Y|Z))^2 = \arg\min E(Y - g(Z))^2$. So, what is $E(Y|Z)$? Take $Y = Y_{n+1}$, Z as previous data ($D = (x_i, y_i)$), and consider models $M_1, \ldots, M_K$.
- Bayesian approach originates with Geisser (1965), Learner (1978)
- Consider $p(y_{n+1}|D) = \sum_{k=1}^{K} p(y_{n+1}|M_k, D)W(M_k|D)$ where $W(M_k|D) = p(D|M_k)W(M_k) / \sum_{k=1}^{K} p(D|M_k)W(M_k)$, $p(D|M_k) = \int p(D_k\theta_k, M_k w(\theta_k|M_k)d\theta_k$ and $p(y_{n+1}|M_k, D)$ is the predictive from model k.

What is an 'Ensemble'?

- Then $E(Y_{n+1}(x)|D) = \sum_{k=1}^{K} \hat{Y}_k(x) W(M_k|D)$ where $Y_k(x) = E(Y(x)|D, M_k)$. AND THIS IS L2 OPTIMAL!
- There are other model averaging procedures with different optimality properties.
- Bagging (Bremen 1984) Bootstrap sample your data, get predictor from each bootstrap sample, vote or take average.
- Boosting (Shapire 1990) Built set of predictors constructed by raising cost where misclassify. Based on group of weak learners.
- Stacking (Wolpert 1992) Use cross-validation to find model weights in model average. Performs well if high model uncertainty.

Facts about ensembles

- Usual procedure is to form a model and then make predictions. Problems stem from model mis-specification (Draper 1995). This means we want a model we can understand and then we see how well it does.
- Ensemble methods reverse this: We find a procedure that predicts well (and hopefully doesn't fit too well or we risk poor generalization error). Then figure out what it means.
- M -closed, M -complete, M -open problems
- You can only beat model average predictions by predictions from a simple model that really is true. A rare case.

Bayes model average

- **Idea**. We have K models, $M_k = \{p(\cdot|\theta_k, M_k)\}$ indexed by k with parameters θ_k having within-model priors $w(\theta_k | M_k)$ and an across model prior $p(M_k)$.
- This is a single hierarchical model with density $p(y, \theta_k, M_k) = p(y|\theta_k, M_k)p(\theta_k|M_k)p(M_k)$ and so satisfies the 'containment principle' of Bayesian statistics (see Seidenfeld; www.reasoner.org).
- Now, the laws of probability can be applied to get expressions for objects of inference.

Model probabilities, etc.

- The marginal experiment is $p(y, M_k)$ – leave out the θ_k's.
- The marginal likelihood of M_k is

$$p(y|M_k) = \int p(y|\theta, M_k)p(\theta_k|M_k)d\theta_k.$$

- The posterior model probabilities are

$$p(M_k|y) = p(y|\theta, M_k)p(\theta_k|M_k) / \sum_k p(y|\theta, M_k)p(\theta_k|M_k).$$

- Now, suppose we want to predict Y_{new} a future observation. So, our model is $p(y, M_k, Y_{new})$.
- We look at $p(Y_{new}|y, M_k)$ and obtain $p(Y_{new}|y)$ by averaging over the densities for Y_{new} from the M_k's.

Bayes model average prediction

- The average of models leads to an average of point predictors by taking expectations. (This is L^2 optimal.)
- Just apply E everywhere:

$$\hat{Y}_{new} = E(Y_{new}|y) = \sum_k E(Y_{new}|M_k, y)p(M_k|y).$$

- Hardy souls can also find an expression for $\text{Var}(Y_{new}|y)$.
- In BMA, can also do model selection...Write

$$\frac{p(M_k|y)}{p(M_j|y)} = \frac{p(y|M_k)\,p(M_k)}{p(y|M_j)\,p(M_j)} = B(k;j)\frac{p(M_k)}{p(M_j)}$$

- The Bayes factor $B(k;j)$ summarizes the support of the data for model k versus model j.

Bayes model average prediction

- For a Bayesian, model uncertainty is now a vector: $(p(M_1|y), \ldots, p(M_K|y))$.
- We could just choose $k = \arg\max_k p(M_k|y)$ and this would amount to the modal model or the BIC....but remember Geisser's 'Theorem': The average is a better predictor than the predictor from any model selected. (Unless one of the models has $W(M_k) = 1$.)
- Usual representation for a class of models is $Y = \mathbf{1}\beta_0 + X_\gamma\beta_\gamma + E$ where γ is a string of 1's and 0's indicating whether or not a given X_k is in the model.
- There are 2^K linear models, must put a prior on the γ and the β's (conditional on the γ). Lots of work on this... (Hoeting et al. 1999; Clyde and George 2004)

Bagging in general

- Bagging is a contraction of bootstrap aggregation, a strategy to improve the predictive accuracy of a model.
- Given a sample, fit a model $\hat{f}(x)$ called the base and then consider predicting the response for a new x_{new}.
- A bagged predictor for x_{new} is found by drawing B bootstrap samples from the training data; i.e., draw B samples of size n with replacement.
- Each sample of size n is used to fit a model $\hat{f}(x)$ so that

$$\hat{f}_{bag}(x^{new}) = \frac{1}{B}\sum_{i=1}^{B} \hat{f}^{i}(x^{new})$$

is the bagged prediction.

- As a generality, bagging can improve good but unstable procedures to make them close(r) to optimal.

sub-bagging

- Buhlmann and Yu (2002) show that $\hat{\sigma}$ has convergence rate $n^{-1/3}$ for smooth regression functions but a non-Gaussian limit distribution.
- Sub-bagging: Rather than choose B bootstrap samples of size n, choose B bootstrap samples of size $m < n$. Then do the same averaging.
- What value of m to choose? Generally good choice is $m = n/2$.
- Bagging doesn't seem to help MARS or other smooth base procedures. Can even make them worse.
- Can use median (trimmed mean?) of B samples in place of mean....may help.

Stacking in general

- Suppose we have K distinct models $f_1,\ldots,f_K$ in which each model has one or more real-valued parameters that must be estimated.
- When plug-in estimators for the parameters in f_k are used, write $f_k(x) = f_k(x|\hat{\theta}_k)$ for the model used to get predictions.
- The task is to find empirical weights $\hat{w}_k$ for the $\hat{f}_k$'s from the training data and then form the stacking prediction at a point x,

$$\hat{f}_{stack}(x) = \sum_{k=1}^{K} \hat{w}_k \hat{f}_k(x).$$

- Let $f_k^{(-i)}(x)$ be the prediction at x using model k, as estimated from training data with the ith observation removed. Then the estimated weight vector $\hat{w} = (\hat{w}_1,\ldots,\hat{w}_K)$ solves

$$\hat{w} = \arg\min_{w} \sum_{i=1}^{n} \Big| y_i - \sum_{k=1}^{K} w_k f_k^{(-i)}(x_i) \Big|^2.$$

This puts low weight on models that have poor leave-one-out CV accuracy in the training sample.

Stacking vs. BMA and bagging

- Stacking is an adaptation of cross-validation to model averaging because the models in the stacking average are weighted by coefficients derived from CV.
- As in BMA, the coefficient of a model is sensitive to how well the model fits the response. However, the BMA coefficients represent model plausibility, a concept related to, but different from, fit.
- In contrast to bagging, stacking puts weights on models rather than pooling over repeated evaluations.
- Loosely, stacking is a version of BMA where the weights use priors that downweight complex or ill-fitting models or a version of bagging with a few highly plausible models rather than full bootstrapping.

Boosting in general

- Idea. Form a classifier h_0. Then improve it by averaging it with another carefully constructed classifier h_1 or series of classifiers $h_1, \ldots, h_K$.
- Construct h_1 by finding the classifier that minimizes the empirical risk of the places (x_i values) where h_0 was wrong.
- The cost at a point where $h_0(x_i) = 0$ when it should have been 1 is exponentially large relative to other x_i's.
- Weights assigned to h_0, h_1, ... decrease as the classifier improves.
- Adaboost algorithm (Freund and Shapire 1999)

Boosting for classification

- Begin with data $(x_1, y_1), \ldots, (x_n, y_n)$, and $y_i \in \{1, -1\}$ and plan T iterations. Distribution of misclassification cost $D_t = (D_t(1), \ldots, D_t(n))$ is initialized at $D(0) = (1/n, \ldots, 1/n)$.
- At each iteration misclassification error is $\bar{E} = P_{D_t}(h_t(X_i) \neq Y_i) = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$.
- Set

$$\alpha_t = \frac{1}{2} \log \frac{1 - \bar{E}}{\bar{E}},$$

and update D_t to D_{t+1} by

$$D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{C_t},$$

in which C_t is a normalization factor to ensure D_{t+1} is a probability vector (of length n).

- The exponential upweights the cost of misclassifications and downweights the cost of correct classifications.

Boosting for Classification

- Set $h^*_{t+1}(x)$ to be

$$h^*_{t+1}(x) = \arg \min_{g \in G} \sum_{i=1}^{n} D_t(i) 1_{\{y_i = g(x_i)\}}.$$

With each iteration, add h^*_{t+1} to a growing sum.

- The updated weighted-majority vote classifier is

$$h_{t+1}(x) = \text{sign}\left(\sum_{s=0}^{t+1} \alpha^s h^*_s(x) \right),$$

Final classifier, h_T, is the boosted version of h_0.

- Lots of work: Error bounds, practical comparisons etc.

Pompous overgeneralizing

- Model averaging beats model selection. BMA is probably optimal in *M*-closed problems.
- Boosting is basically additive logistic regression. It can be lead astray under random error (Mease 2008)
- Stacking is great for averaging over different model types
- Bagging trees will become 'standard' for classification.
- SVM and RVM will be important for *M*-open problems. Both give sparsity, but stability?

"Let's solve these first. We can worry about data mining later."

How is Big Data Different? A Paradigm Shift

Jennifer Clarke, Ph.D.

Associate Professor
Department of Statistics
Department of Food Science and Technology
University of Nebraska Lincoln

ASA Snake River Chapter, Meridian, ID, May 29 2015

Outline

Shift 1. Trickle to Firehose

Shift 2. Experiment to Observation?

Shift 3. Low Dimension ($n > p$) to High Dimension ($n < p$)

Shift 4. Modeling to Prediction

Statistics is important

Trickle to Firehose

- As statisticians we enjoy working with data - preprocessing, visualizing, modeling, inference, analysis, sharing, etc. These activities become difficult or impossible when the amount of data is large.
- Each piece of analysis requires considerable time and effort, and consideration of computational expense
- What do we do when data is so large it can't fit on one computer? What about 'garbage in, garbage out'?
- We rethink data as dynamic and emphasize random sampling

Trickle to Firehose

- Think about distributed processing. Can analysis be done in parallel or online? Hadoop, MapReduce
- Data gets bigger, not smaller, during preprocessing so plan ahead
- Computer scientists and system administrators are your friends
- Learn new computational skills (Python, SQL) and learn from others (social media)

Experiment to Observation?

- Much of Big Data is collected without thought. Period. Collecting is easy, analysis is hard.
- Experimental design has played a limited role, but is critical to inference and prediction.
- Sample size may not correlate with data size, e.g., genomics, social networks.
- Need for modern experimental design with detailed data provenance

Experiment to Observation?

- Do we need statistics? YES.
- 'To consult the statistician after an experiment is finished is often merely to ask him to conduct a post-mortem examination. He can perhaps say what the experiment died of.' Fisher
- Still relevant: sampling populations, confounders, multiple testing, bias, and overfitting
- Usually not randomization
- Unstructured data: information that either does not have a pre-defined data model and/or is not organized in a predefined manner. (www.mapr.com)

Low Dimension ($n > p$) to High Dimension ($n < p$)

- Figure out ways to make p smaller: variable selection, variable summarization, variable sampling.
- Variable selection: sure independence screening (Fan and Lv 2008), LASSO (Tibshirani 1996), regularization
- In the context of linear regression $E(y|X = x) = \alpha + \beta x$, estimates are chosen to minimize $\arg\min_{\beta} \sum (y_i - \alpha - \beta x_i)^2$ subject to $\sum |\beta_j| < t$. This is equivalent to minimizing $\frac{1}{2n} \sum (y_i - \alpha - \beta x_i)^2 + \lambda \sum |\beta_j|$.
- one can summarize variables by PCA, cluster medioids, etc.

Low Dimension ($n > p$) to High Dimension ($n < p$)

- Variable sampling (huh?). Can model subsets of data where subsets are random samples of observations AND variables.
- Overfitting is a BIG problem.
- Correct for multiple testing ... FDR, Westfall-Young
- Problem drives solution ... don't 'hit all the nails'

Shift 4. Modeling to Prediction

- Idea: When data is unstructured or otherwise complex, model uncertainty is high
- If the goal is prediction, average or ensemble. Think bagging, boosting. This will reduce variability without increasing bias (while accounting for model uncertainty).
- Focus on accurate prediction and sequential/prequential analysis (interactive)
- Approach inference by assessing and dissecting predictor
- Uncertainty and variability based on random sampling and/or permutation
- Error rate depends on correlation between trees and strength of individual trees

Shift 4. Modeling to Prediction

- Linear models are like donkeys. Treat them right and they'll carry you, grudgingly, as far as they can. They will bray when they are unhappy but you will know how to feed and water them. They don't travel far.
- Neural Networks are like a big pile of snakes. Some are poisonous though most aren't. There are different sizes and colors. You can grab one that looks right but since you can't see the whole thing it will coil around in unexpected ways.
- Trees are like foxes. They're clever and run around all over the place. Catch the ones that will solve your problem.
- Ensemble methods are packrats. They always have something that works well but it may be a mess. This can save time catching snakes, chasing foxes, or beating donkeys. But, you don't know what you've really got.

Random Forests: Variable Importance

- If scores independent across trees, can normalize and determine significance.
- Local variable importance scores can be calculated *for each case* (Strobl and Zeileis 2008).

Random Forests: Proximities

- Proximities provide a nice way to visualize observations
- Once all trees grown get terminal node for each observation. If obs k and k' in same node increase proximity by one. Add across trees and scale by number of trees. Yield N by N proximity matrix of $prox(k,k')$ for all k,k'.
- Note $(1 - prox(k,k'))$ is a distance measure; project distances into lower dimensional space using Multidimensional Scaling (Eigenvectors and Eigenvalues).

Seeing the Forest for the Trees

Jennifer Clarke, Ph.D.

Associate Professor
Department of Statistics
Department of Food Science and Technology
University of Nebraska Lincoln

ASA Snake River Chapter, Meridian, ID, May 29 2015

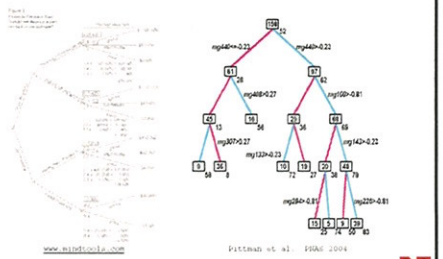
What are 'Random (Decision) Forests'?

- A random forest is an ensemble classifier that consists of many decision trees.
- The idea of an ensemble is to learn a set of classifiers and combine their predictions. The output can be the modal class or a weighted average of the tree classifications.
- The motivation of ensemble classifiers is to reduce variance (less dependent on peculiarities of single training set) and reduce bias (learn more expressive concept class)

Decision Trees

- Classification trees are used to predict membership of subjects in classes of a categorical dependent variable from measurements on one or more predictor variables.
- Very flexible and rich model class, good as exploratory technique, for high dimensional data, and/or when traditional methods (Discriminant Analysis, Nonlinear Estimation) inadequate
- Nonlinear and hierarchical, focus on interactions
- Original data set is progressively split into mutually exclusive regions using a series of binary splits

It Can Be Done ... Decision Trees



Outline

- Classification Tree
- Random Forests
- Consistency and Sparsity
- Discussion

Classification Tree

- Goal is to create a model that predicts the class to which an observation belongs based on several input variables.
- Each interior node corresponds to an explanatory variable; edges link parent nodes to children nodes. Each leaf represents a subset of the explanatory variable space described by the path from the root to the leaf.
- A tree can be learned by choosing a variable at each step that is the best variable to use in splitting the set of observations, where 'best' is defined by how well the variable splits the observations into homogeneous subgroups.
- There are different metrics for what is 'best', including Gini impurity (CART), information gain (ID3, C4.5), and Bayes' factors (Clarke et al. 2008; Chipman et al. 2010).

Classification Tree

Advantages

- Can handle both categorical and continuous data
- Robust to distributional assumptions
- Perform well in high dimensions

Disadvantages

- Learning optimal tree is NP-complete, i.e., 'Find the binary decision tree which minimizes the expected number of tests required to identify the unknown object'(Hyafil and Rivest 1976)
- Can create over-complex trees that do not generalize, i.e., overfitting
- Some concepts hard to learn, e.g., XOR (see logic trees, Ruczinski et al. 2003)

Impurity Function

- Idea: Measure of variance or purity of region containing data points that may come from different classes. If K classes, then impurity is function of probabilities $p_1, \ldots, p_K$ of any point in the region being in class $k, k = 1, \ldots, K$.
- An impurity function is a function φ defined on the set of all K-tuples of numbers $(p_1, \ldots, p_K)$ satisfying $p_j \geq 0, j = 1, \ldots, K$, $\sum_j p_j = 1$ with the properties:
  1. φ is maximum when p_js are all equal
  2. φ is minimum when each K-tuple has a single $p_j = 1$ and all other $p_j = 0$
  3. φ is symmetric function of the p_js
- For given node t impurity measure is $\varphi(p(1 \mid t), p(2 \mid t), \ldots, p(K \mid t))$.

Random Forests (Brieman and Cutler)

- Idea: Learn many classification trees, get the classification from each tree (vote), and classify an object by the modal classification (class with the most votes)
- Growth:
  1. Sample a training set of n cases at random with replacement, $n \approx (2/3)N$.
  2. Select $m \ll d$; at each node m input variables are selected at random as candidate split variables
  3. No pruning
- Error rate depends on correlation between trees and strength of individual trees

Random Forests: Out-of-bag (OOB) Error

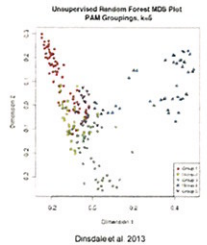
- Out-of-bag (OOB) error estimate is unbiased estimate of test set error
- Each tree constructed with different bootstrap sample, leaving out 1/3 of observations
- For each left out sample get predictions from left out trees; proportion of incorrect predictions averaged over all left out cases is oob estimate

Random Forests: Variable Importance

- The importance of variable x is determined by getting predictions for oob cases, count number of correct votes, then randomly permute values of x in oob cases and repeat. Difference in number of correct votes from two procedures is raw variable importance score for x.
- If scores independent across trees, can normalize and determine significance.
- Local variable importance scores can be calculated *for each case* (Strobl and Zeileis 2008).

Random Forests: Proximities

- Proximities provide a nice way to visualize observations
- Once all trees grown get terminal node for each observation. If obs k and k' in same node increase proximity by one. Add across trees and scale by number of trees. Yield N by N proximity matrix of $prox(k, k')$ for all k, k'.
- Note $(1 - prox(k, k'))$ is a distance measure; project distances into lower dimensional space using Multidimensional Scaling (Eigenvectors and Eigenvalues).



Dinsdale et al. 2013

Consistency

- Lin and Jeon (JASA 2006) one of the first papers to try to explain RF; relate to adaptive nearest neighbors (TR from 2002 available from U of Wisconsin)
- Breiman (TR 2004) shows consistency of simplified RF algorithm; key to consistency is random selection of splitting variables at nodes
- Meinshausen (JMLR 2006) proves consistency of quantile regression RF
- Most recent contribution by Gérard Biau (2008, 2012, 2015) on the consistency of RF and averaging classifiers

Biau: Consistency of RFs

- Assume randomization independent of data, k_n terminal nodes, midpoint splits, variable j has probability p_{nj} of being selected where $p_{nj} \in (0, 1)$ tends to $1/S$ for $j \in S$ as $n \to \infty$
- Variance of forests estimate is smaller than the variance of a single tree estimate, i.e., $O(k_n/n(\log k_n)^{(S/2d)})$ vs. $O(k_n/n)$
- Bias decreases to 0 at a rate depending on S, not d
- With the optimal choice of $k_n \propto n^{(1/(1+(0.75/S\log 2)))}$, RF estimate is consistent at rate $O(n^{-0.75/(S\log 2+0.75)})$
- This rate is strictly faster than the usual rate $n^{-2/(d+2)}$ as soon as $S < /0.54dj$

Biau: Consistency and Sparsity

Conclusions:

- RF is consistent
- RF rate of convergence is faster than usual rate when $S < \lfloor 0.54dj \rfloor$
- Rate of convergence depends only on the number of strong features S and not on the number of noise variables present
- RF adapts to sparsity

Conclusion

- Tree-based methods are primarily an automated machine learning technique. There is growing interest in applying tree-based methods in biomedical applications, partly due to the rising challenges in analyzing genomic data with a large number of predictors and a far smaller number of observations
- The main advantage of tree-based methods is their flexibility and intuitive structures. However, because of their adaptive nature, statistical inference based on tree-based methodology is generally difficult.
- Bayesian approaches may offer another way to construct forests by including trees with a certain level of posterior probability (Clarke et al. 2008; Chipman et al. 2010)

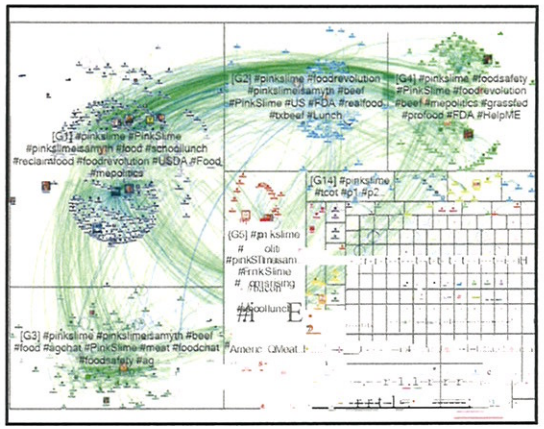
Statistical Networks and Social Media

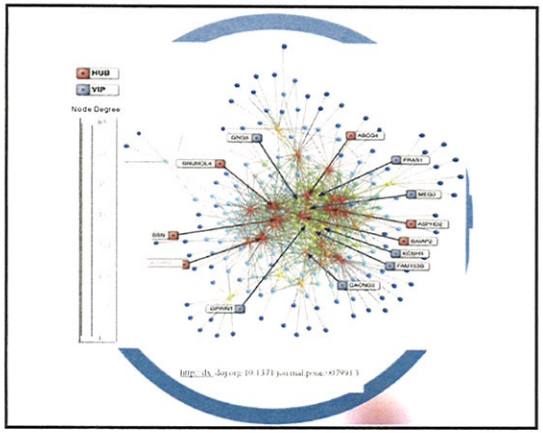
Jennifer Clarke

ref: Eric Kolaczyk , Boston University

Why network analysis?

The analysis of networks has gained tremendous interest with the growth of big data and social media





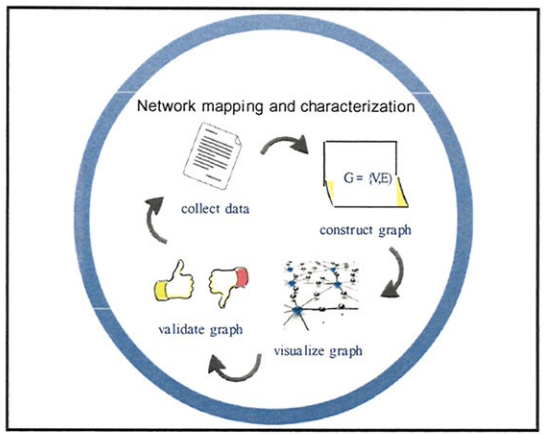
We will focus on statistical analysis of networks, i.e., analysis of measurements conceptualized as a network

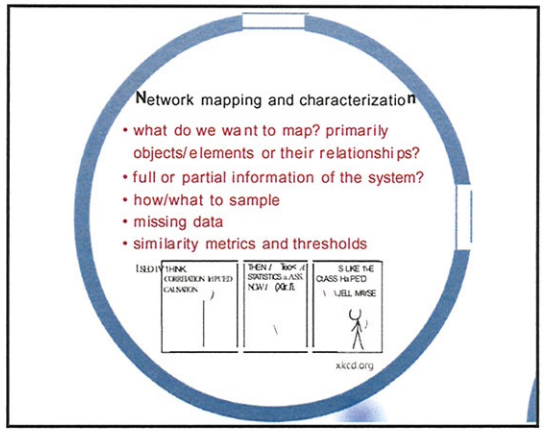
Although networks can be from different contexts (social, technological, biological) there are certain canonical problems that we try to solve

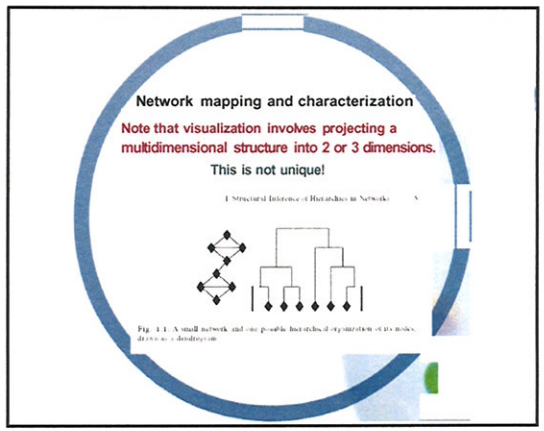
- network mapping and characterization
- network sampling
- network inference

[network processes (dynamic)]

Chalenges: relational, complex dependencies, high dimensional, dynamic


Network mapping and characterization
G = (V,E)
collect data
construct graph
validate graph
visualize graph


Network mapping and characterization
• what do we want to map? primarily objects/elements or their relationships?
• full or partial information of the system?
• how/what to sample
• missing data
• similarity metrics and thresholds
xkcd.org


Network mapping and characterization
Note that visualization involves projecting a multidimensional structure into 2 or 3 dimensions.
This is not unique!

Network mapping and characterization

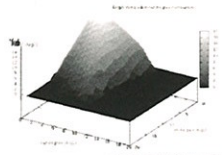
Characterization is based on structural properties. These include

- degree distribution (connections)
- vertex/edge centrality (hub/spokes)
- community structure (topology)

There are a gazillion ways to measure each of these, so we'll consider just a few here.

Degree distribution

- the probability distribution p of the number of connections each node has to other nodes across the entire network (degree). common choices for p are binomial and power Law.
- if edges are directed, in-degree and out-degree



component structure

- network scientists are interested in components and cliques, i.e., sub-sets in which nodes are more closely and intensely tied relative to other members of the network
- average clustering coefficient: $C(v)$ or fraction of pairs of nodes (u,w) such that (u,w) are adjacent to v and share edges with v

network sampling

- the observed network is an example of a true 'underlying' network, or network is too Large to process as a whole
- includes various approaches: direct, snowball, random walk, importance, MCMC, Gibbs

- Task 1: sample V or E and estimate average degree or degree distribution
- Task 2: sample small subnetworks while maintaining global structural characteristics
- Task 3: sample local substructures (ike cliques) and estimate their relative frequencies

network sampling

- assume network is connected, and if not, we can ignore isolated nodes.
- assume network is hidden but allows crawling, i.e., can explore neighbors of a given node.

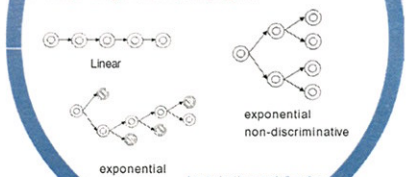
Graph traversal. sampling without replacement

snowball sampling: Like subject referral. This sampling technique works Like chain referral. After observing the initial subject, the researcher asks for assistance from the subject to help identify people with a simi ar trait of interest.

snowball sampling (SBS)

advantages: good for rare subpopulations; cheap and efficient; requires Little planning.

Disadvantages: little researcher control; representativeness of sample not guaranteed; sampling bias



network sampling

- assume network is connected, and if not, we can ignore isolated nodes.
- assume network is hidden but allows crawling, i.e., can explore neighbors of a given node.

Random walk techniques. sampling with replacement

classic random walk sampling: at each iteration one of neighbors of current node is selected to visit.

p_u,v = 1/d(u) if v is in adj(u) and zero otherwise

random walk sampling

advantages: samples each edge uniformly

disadvantages: is biased towards nodes with many edges

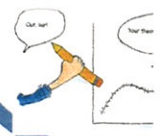
P(sample node u) d(u)/2m

where m is number of edges in graph

Can address bias with added complexity, M-H correction

define task | select sampling objective | data access restrictions?

choose sampling method | evaluate

Social Media

- NodeXL (Network Overview Discovery and Exploration for Excel) http://www.smrfoundation.org/nodexl/
- cuttlefish (network visualization) http://cuttlefish.sourceforge.net/
- graph-tool (python toolbox for analysis) https://graph-tool.skewed.de/