The Computational Life of an Applied Mathematician

with personal examples from data mining and time series analysis

Alan Ableson

Background

- Undergraduate studies in applied math
  - operations research
  - numerical methods
- Mathematical and computational problems presented in clear “Problem -> Solution” format

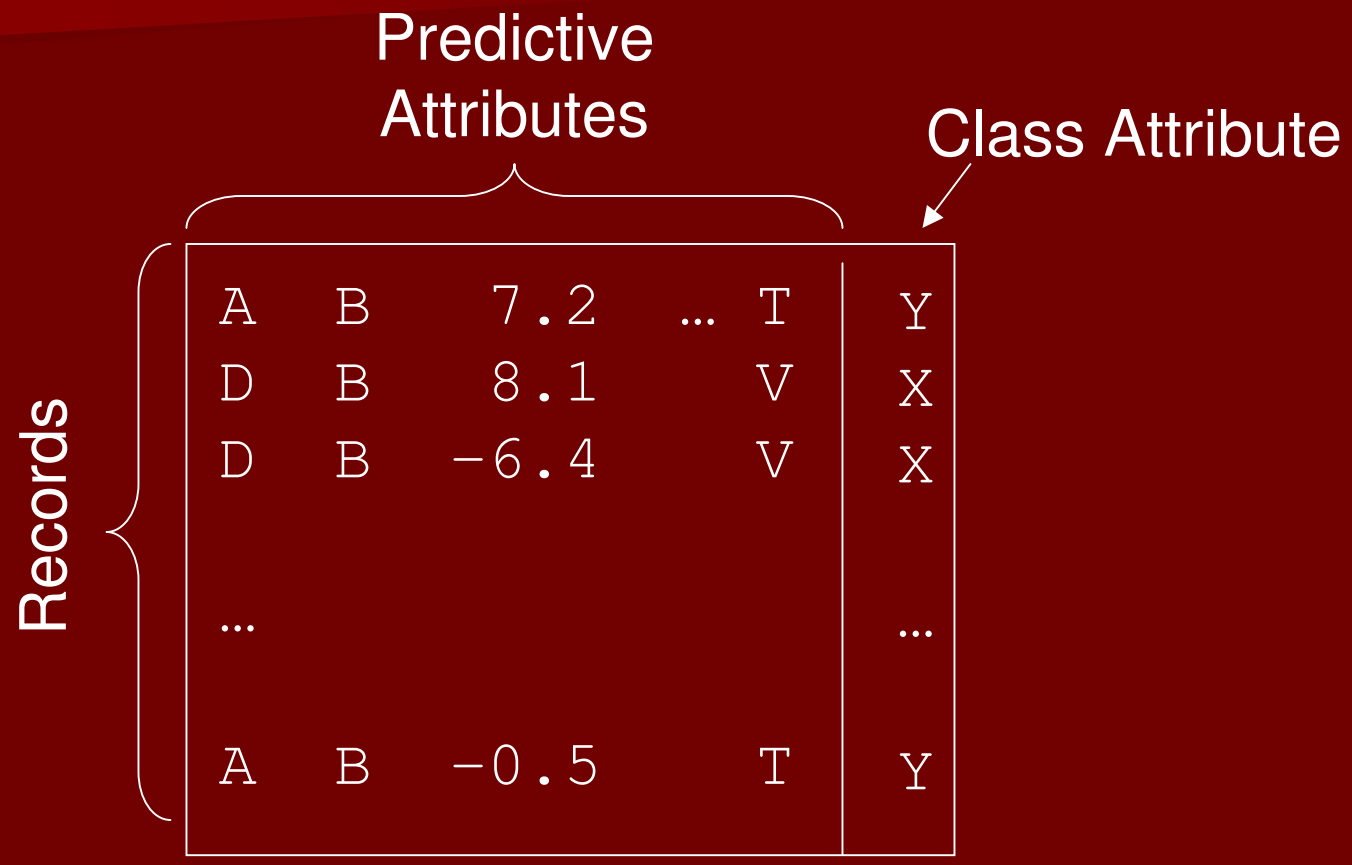
Background

- Graduate studies and corporate work revealed limitations to that approach!
  - Research areas have independent history of both key questions in the field **and** standard mathematical/computational approaches
  - Key questions are (hopefully!) relevant and well-posed
  - Corresponding mathematical/computational formulations…?

Background

- Present two problems areas
  - Data mining in tabular data
  - Time series analysis
- Basic problem description/motivation
- Focus on computational aspect
- Limitations in domain-specific approach

Example 1: Classification Paradigm For Tabular Data



Data Table

Uses of Classification in Research

- Classification has become a key module in scientific discovery streams
- Some examples:
  - **Document recognition**: Image bitmap attributes predicting image type
  - **Biomedicine**: arm motion time series attributes to distinguish stroke / non-stroke victims
  - **Protein Structure Prediction**: amalgamating scores from simple structure evaluators into a global prediction

Interpretability of Classifiers is Important

- Making a classification module more accurate through better understanding of data is more useful than through tweaking learner parameters
- Interpretable classifiers provide *proof by construction* of the predictive relationships
- Interpretable classifiers put the focus of classification on the *data* instead of the *learner*

Logistic Discriminants

- Logistic discriminants are interpretable
  - Simple log-linear form
  - Single parameter associated with each (attribute value + class value) pair
  - Parameter reflects odds of class value being associated with attribute value
  - Surprisingly accurate in general!
- Not always as accurate as non-linear classifiers
  - Linear form
  - Single attribute values*

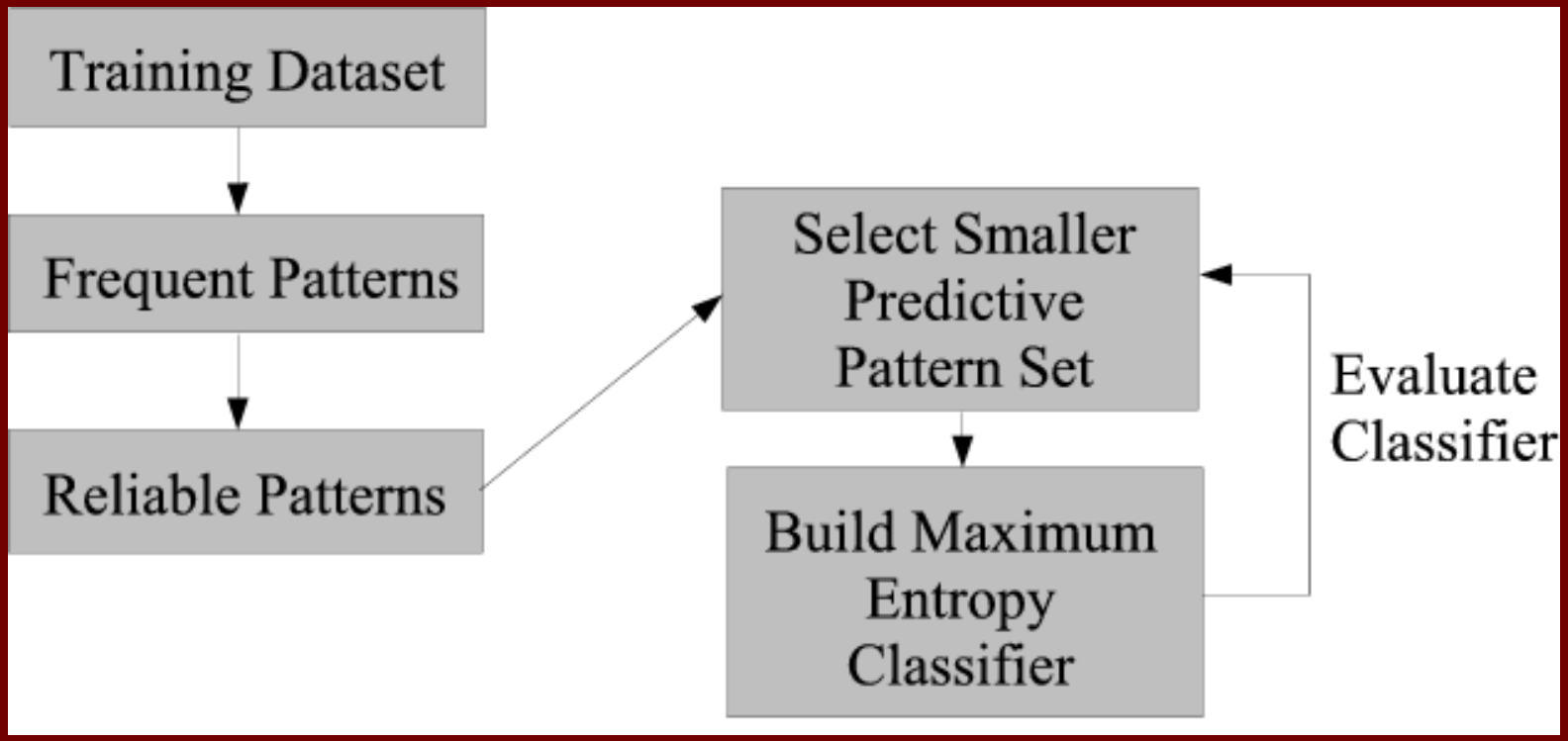
Example of Logistic Log Odds

	Low Income	High Income
education = Preschool	-	-8.6
education = 9th	-	-0.3
education = 12th	-	0.2
education = HS-grad	-	1.0
education = Assoc (academic)	-	1.5
education = Some-college	-	1.2
education = Bachelors	-	2.3
education = Prof-school	-	3.6
education = Masters	-	2.9
education = Doctorate	-	3.7

Proposed Structure for Pattern-Based Classifiers

- Extension to Logistic Discriminant
  - Use *multi-attribute patterns* instead of single attribute values
  - Keep simple log-linear form
  - Single parameter associated with each (pattern + class value) pair
  - Parameter reflects odds of pattern being associated with class value
  - Include as broad a pattern search as possible
  - Try to limit patterns to interpretable sizes

Proposal for Pattern-Based Classifier Using Maximum Entropy Distributions



Example

Training Dataset

Age	Sex	Education		Income
< 40	M	up to HS	…	Low
> 40	F	above HS		High
…				

Frequent Patterns

Age<40 AND Income = Low
Age<40 AND Sex = M AND Income = Low
Age<40 AND Sex=M AND Education = Bachelors AND Income = Low
…

up to 100,000 patterns per class

Example (cont.)

Reliable Patterns

Age < 40 AND Income = Low
Marital Status = Married AND Income = High
Marital Status = Married AND Education = up to HS AND Income = Low
Marital Status = Married AND Occupation = Prof/Specialty AND Income = High
…

up to 5,000 patterns per class

Weighted Patterns making Classifier

	High Income
Marital St's=Married AND Education=above HS	1.9
Marital St's=Married AND Occupation=Service	-1.2
Education=Doctorate	2.2
Claimed capital gains=True	1.9
…	

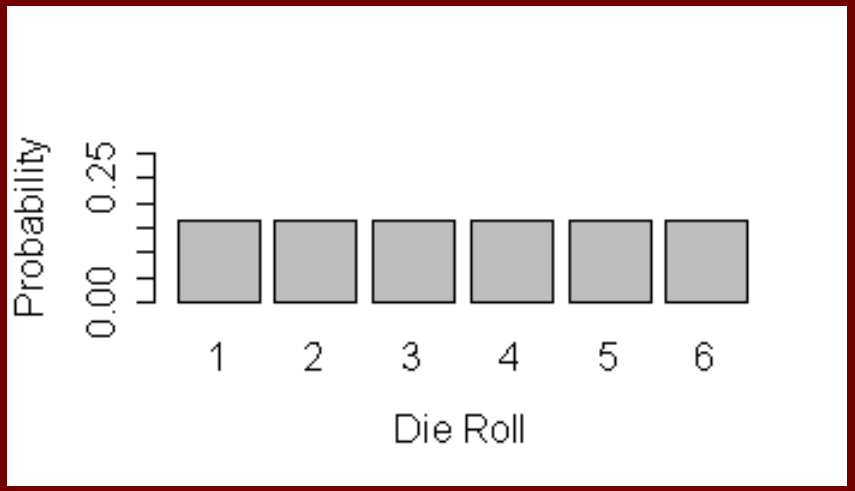
up to 50 patterns total

Computational Challenge

- Given a set of patterns, and their relative frequency within each class, how should we assign their relative odds?
- Wanted to use probability distributions for generality
- Wanted to use only the selected patterns in computing the probabilities
- Wanted a log-linear model to relate model parameters directly to the odds
- Maximum Entropy Distributions seemed like the most appropriate model

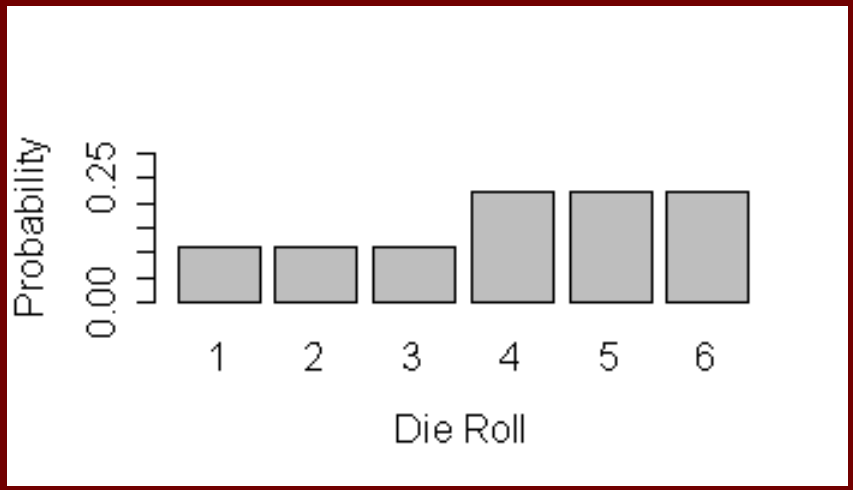
Maximum Entropy Distributions

- If shown a 6 sided die, and asked “What is the probability of rolling a 1?” what is the most “reasonable” answer?
- Any answer but 1/6 requires assuming information not in evidence
- Mathematically, 1/6 gives the most uniform distribution, subject to P(1,2,..,6) = 1



Maximum Entropy Distributions

- Add the statement: the die is weighted so probability of rolling 4,5 or 6 is 2/3. What is the probability of rolling a 1?
- Symmetry, or lack of other assumptions, leads to P(1) = 1/9



Maximum Entropy Distributions

- More complicated: consider two dice, knowing
  - P(D1 = 6) = 1/5
  - P(D2 = 6) = 1/5
  - P(D1=6, D2=6) = 1/30
- What is P(D1 = 1)?
- Many distributions satisfy given constraints
- Obtain maximum uniformity by selecting the distribution with *maximum entropy*

Computing Maximum Entropy Distributions

- Computationally, maximum entropy models are found through a dual problem, involving a minimization procedure
- Distributions have one parameter per constraint, simple exponential model
- Search for parameter values such that

  P(constrained states) -> desired constraint probability

  for all constraints
- Exponential form guarantees final distribution will have maximum entropy

Perils of Domain Literature 1

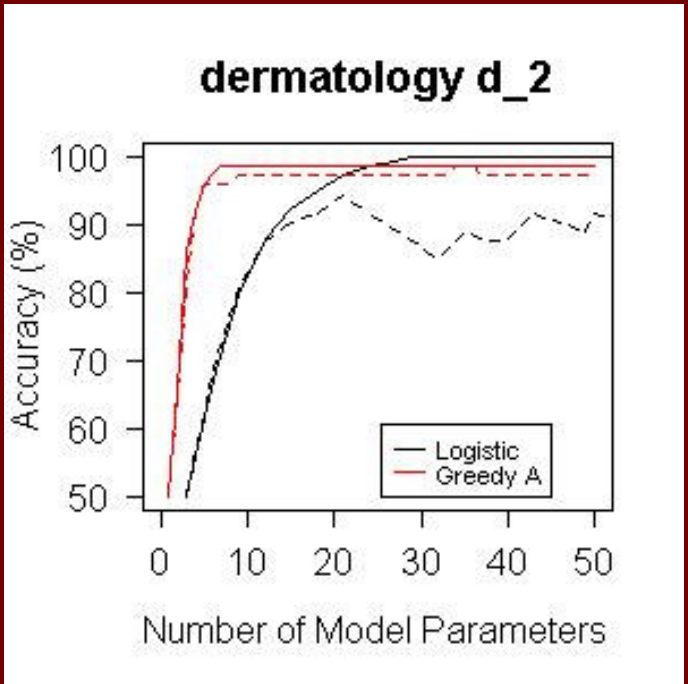
- “How to” aspects of computing often explained in more detail in application areas, rather than in theoretical treatises
- Maximum entropy distributions disproportionately popular in natural language processing
- All major application papers pointed to computational approach in one reference
- Reference outlined theory and detailed computational algorithm, suitable for programming
- (Algorithm name even contained the word “Improved”!)

Resolution

- Implemented the given algorithm, found it fairly slow
- Standard software optimization tricks only gained some performance
- Back to the literature!
- Fortunately, found second, less prominent reference: “A comparison of Algorithms for Maximum Entropy Parameter Estimation” (Malouf 2002)
- Advice: “Improved” way is slow; use off-the-shelf gradient descent algorithm for faster convergence
- Performance gain: min of 10 times faster

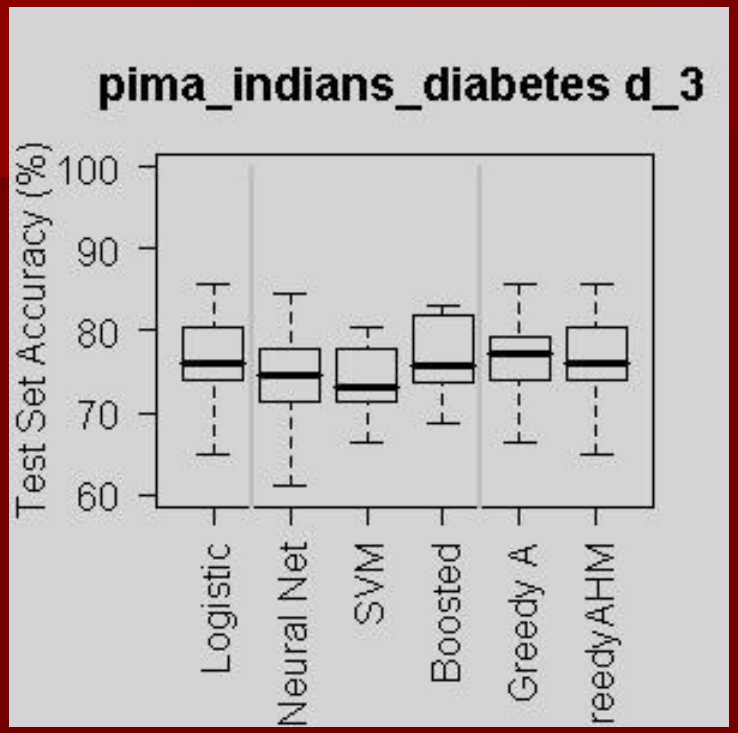
Anecdote I - Dermatology

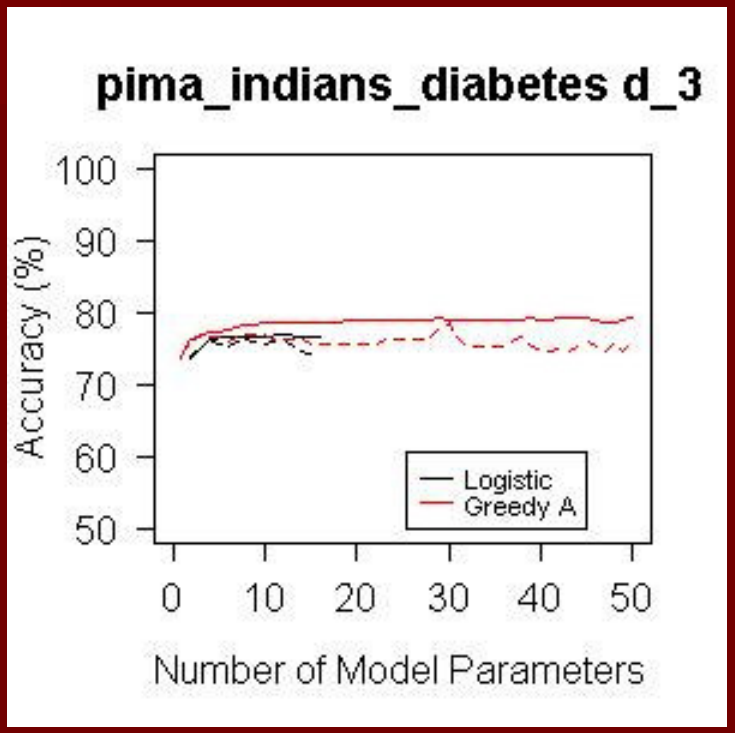
- Goal: predicting underlying skin condition based on non-specific symptoms



- Maximum entropy models used combinations of 2-4 symptoms to correctly identify condition
- Required far fewer patterns than logistic predictor

Anecdote II - Diabetes





	No Diabetes	Diabetes
Glucose Tolerance=Low	--	1.8
BMI=Low	--	-1.5

Example II – Time Series Analysis

- Problem: given position measurements over time, identify frequencies of vibration in signal
- Clearly, the solution involves the Fourier transform, but the details!
- In math, Fourier transforms are usually presented from a PDE or real/complex analysis perspective
- In those approaches, “Data” is never really included in the presentation

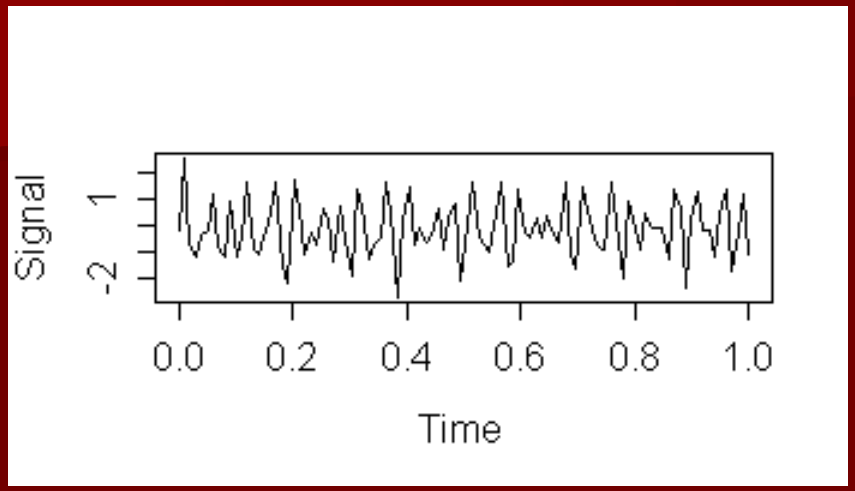
Example II – Time Series Analysis

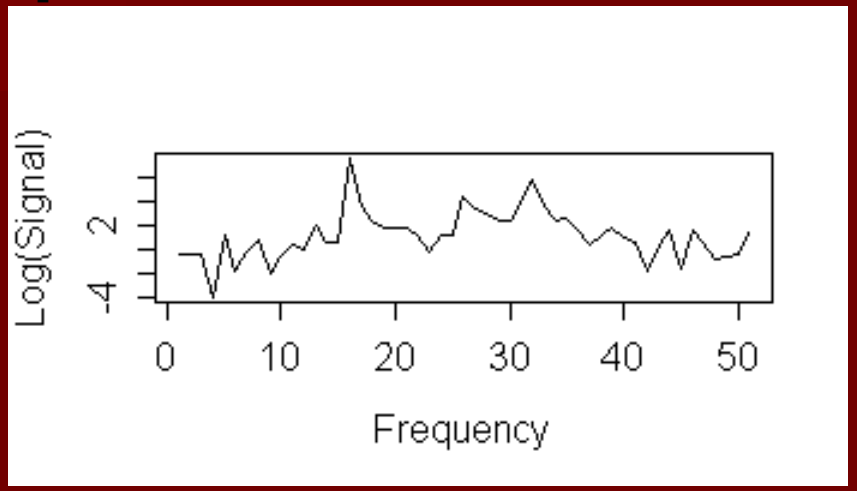
- Continuous and discrete Fourier transforms are often introduced as natural complements to each other ("and ,by the way, if your data happens to be discrete…")
- Unfortunately, DFT/FFT of sampled continuous data is *not* statistically equivalent to the FT of the underlying continuous time series (not widely advertised in many disciplines) (Thomson, 1982)
- Implies computational consequences for spectrum estimation from digitized data

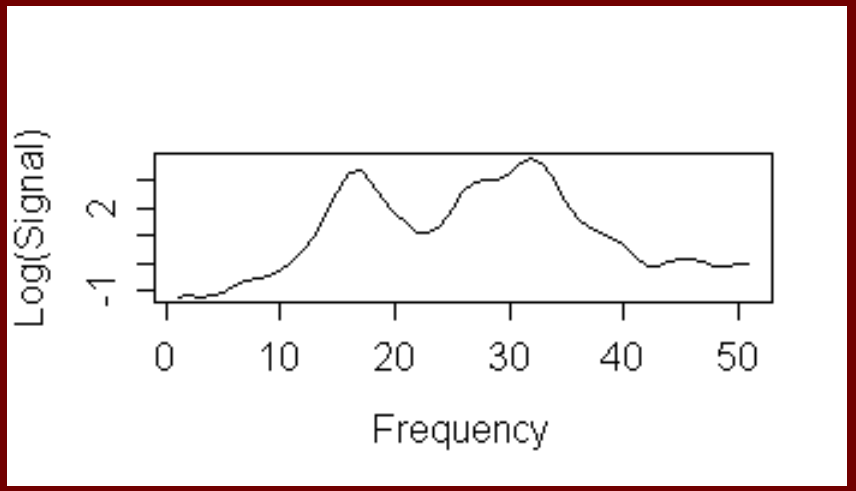
Smoothing Frequency Domain

- Researchers are typically dissatisfied with raw DFT/FFT output (too "spikey")
- Variance between frequencies treated as noise
- Smoothing/averaging approaches developed to discover "true" signal frequencies

Example


Signal
-2
1
0.0
0.2
0.4
0.6
0.8
1.0
Time


Log(Signal)
-4
2
0
10
20
30
40
50
Frequency


Log(Signal)
-1
2
0
10
20
30
40
50
Frequency

Smoothing Design Questions

- In time domain or frequency domain?
- Square windows or tapered windows?
- Window width?
- Constant or data-dependent width?

- What is the scientific question, and what property to do need to correctly/accurately estimate to answer that question?

Perils of Domain Literature 2

- Found in literature:
  - Data-driven smoothing approach
  - Smoothing depends on frequency
  - Smoothing doesn't over-smooth narrow peaks, but does smooth broad peaks
  - Smoothing depends on location of highest peak – (what if there are two similar peaks?)
  - Amount of smoothing depends on user-defined parameters a = 0.22 and b = 3.5. These "worked well"
- I don't know the right answer, but I'm pretty sure that's not it…

Discussion

- Math, statistics, and computation can be found in more and more disciplines every year
- In many of these disciplines, training in math and computation stops after first year calculus
- Students learn discipline-specific computational approaches in-situ from fellow discipline members
- Can lead to same institutionalization of inefficient or inappropriate computational approaches
- Big question for programs at Queen's: is there an approach to teaching math/numerical methods/statistics that would ameliorate this?