Probability and Statistics for Cyber Security

David J. Marchette

June 23, 2018

Topics

Topics

Take-Away Points

- Mathematics and statistics provide many tools for cyber security.
- Simple can be powerful.
  - Complicated models or algorithms are not always necessary.
  - Sometimes they are.
- "Complicated" things become "simple" with familiarity.
- High dimensional data is complicated, messy, and can fool you.
- Know your data!

Two Cultures

*"There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown."**

There are many aspects of this dichotomy:

- Modeling – algorithms.
- Parametric – non-parametric.
- Statistics – machine learning.
- Inference – prediction.**
- "Small" data – "big data".
- "Traditional" statistics – computational statistics.

*Leo Breiman, Statistical Science 2001, Vol. 16, No. 3, 199231

**Donoho, D. (2015, September). 50 years of Data Science. In Princeton NJ, Tukey Centennial Workshop.

`http://www.economicsguy.com/wp-content/uploads/2016/06/50YearsDataScience.pdf` accessed 8/8/2017

The Illusion of Progress

*"... [comparative studies] often fail to take into account important aspects of real problems, so that the apparent superiority of more sophisticated methods may be something of an illusion."**

- Simple models often produce essentially the same accuracy as more complicated models. These can be easier to understand, fit, and may have fewer parameters to choose – possibly resulting in lower variance.
- The data you get is rarely (if ever) a true random draw from the distribution you will be running your trained/implemented algorithm on.
  - This is particularly important in cyber security.
  - By its nature, cyber security data is non-stationary, and today's data may look very different from tomorrow's.

*David Hand, Statistical Science 2006, Vol. 21, No. 1, 114

The Illusion of Progress

- When building a model, one makes assumptions, which are often not testable, and which can impact the ultimate performance.
  - Simpler models (may) have fewer assumptions.
  - Non-parametric (may) be superior to parametric in that they (tend to) make fewer assumptions.
    - However, if the assumptions are true, parametric may be superior.
    - "Good" non-parametric algorithms would be "nearly as good" as the parametric, while allowing a hedge on the assumptions.

Hand suggests we spend less time developing the "next great classifier" and more time on methods that mitigate the above issues.

Topics

Basic Probability

- Probability is the mathematical formalism that encodes uncertainty and allows for computation and analysis.
- Thinking from a frequentist perspective, the probability $P(A)$ of an event A is the proportion of times the event A would occur. This can be made precise, and there are several formalisms using "sample spaces" and measure theory, but the basic intuition is enough for our purposes.
- Probability has the following properties:
  1. $P(\emptyset) = 0$.
  2. $P(\Omega) = 1$.
  3. If $A \cap B = \emptyset$ then $P(A \cup B) = P(A) + P(B)$.
  4. If $A \subset B$ then $P(A) \leq P(B)$ and $P(B \setminus A) = P(B) - P(A)$.
  5. Writing A^c for $\Omega \setminus A$, $P(A^c) = 1 - P(A)$.

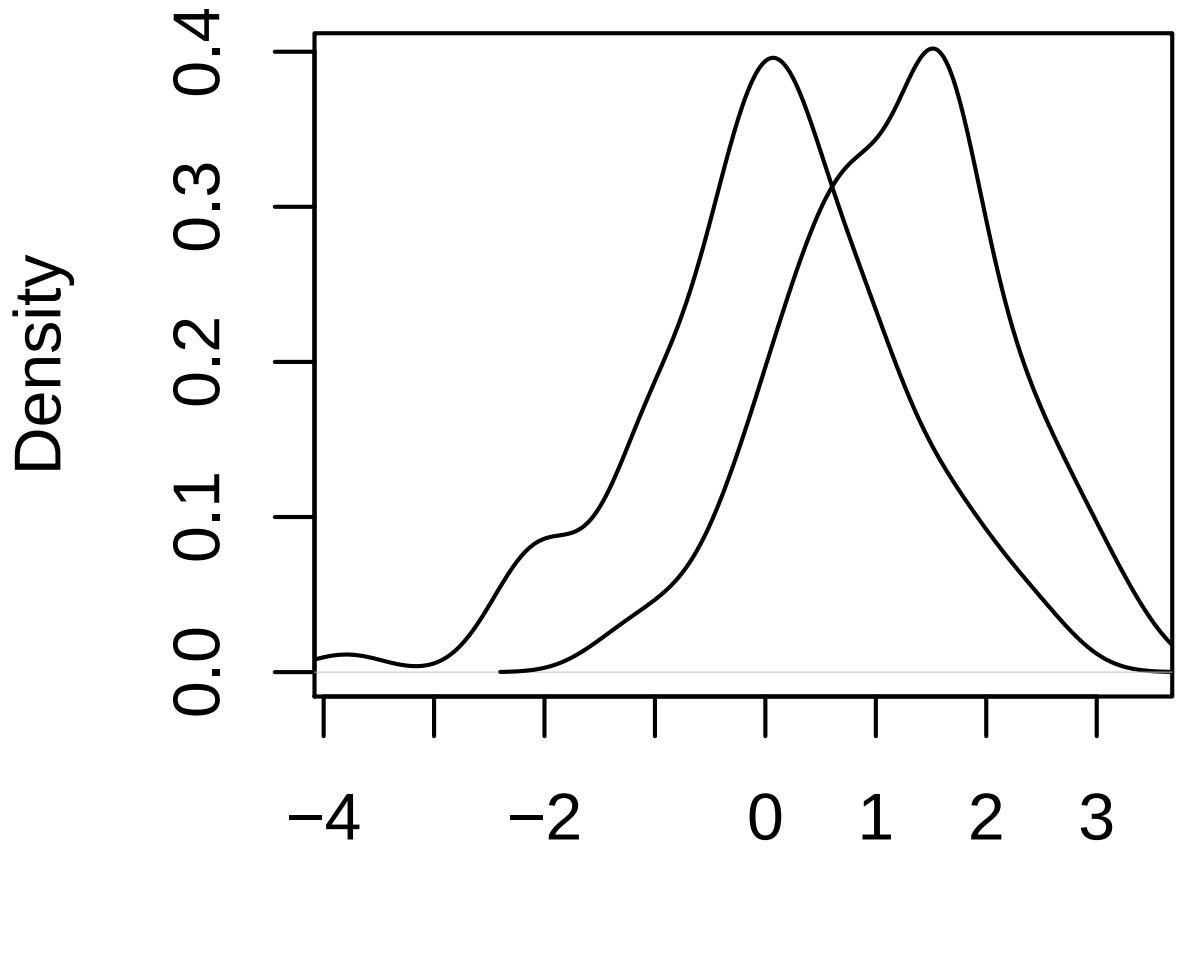
Probability Densities

- A coin has a probability of heads and one of tails and this completely characterizes the "coin flip" distribution (assuming we ignore the probability of it landing on an edge).
- We call this: $P(H) = p, P(T) = 1 - p$ the probability mass function.
- For continuous data – say, normally distributed data, or uniform on $[0, 1]$ – the relevant concept is a probability density function.
- Formally, the density function $f(x)$ satisfies

$$P[x \in A] = \int_A f(x)dx$$

Probability Densities

- Probability densities give us a powerful tool to analyze data.

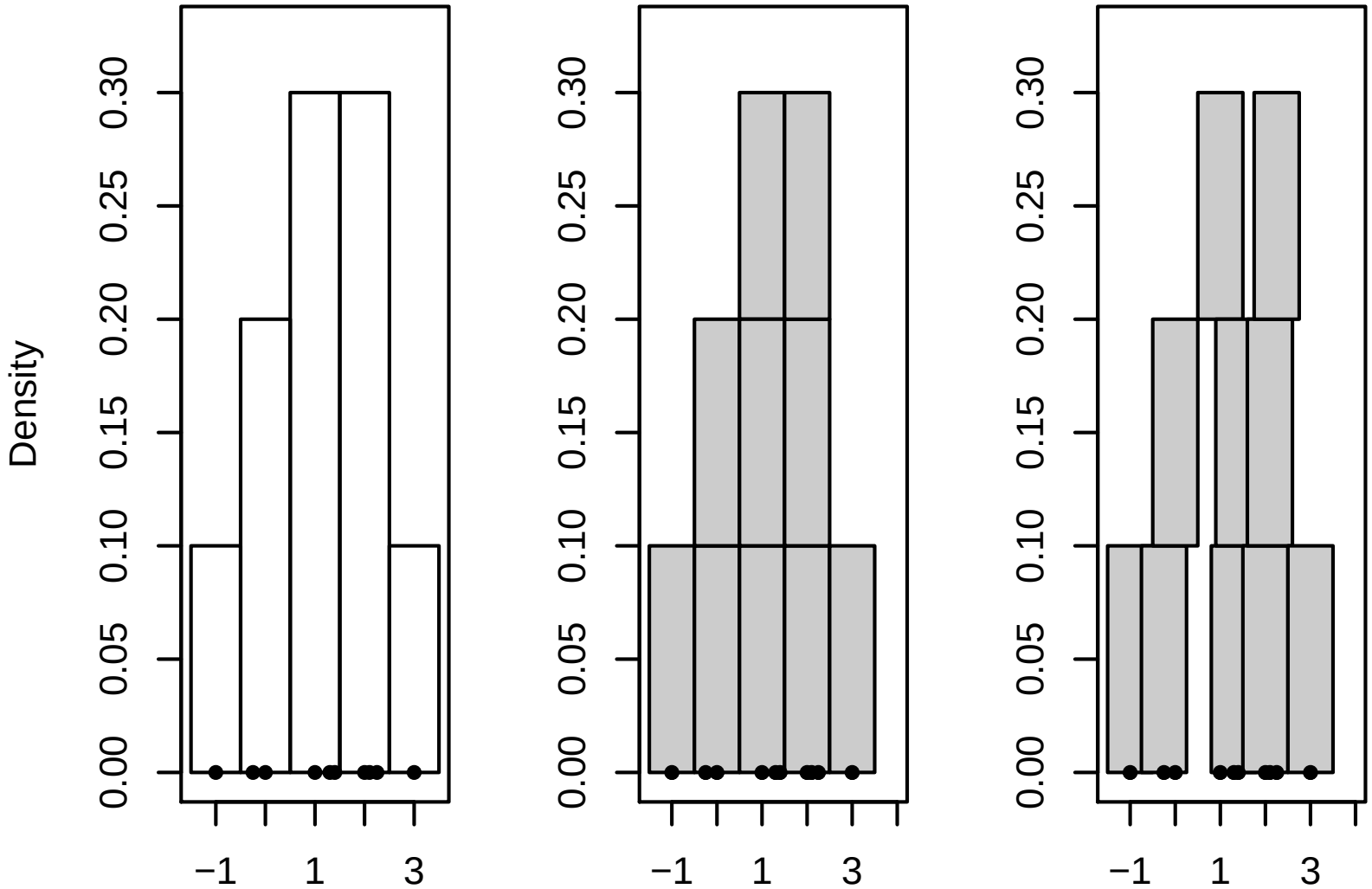


Topics

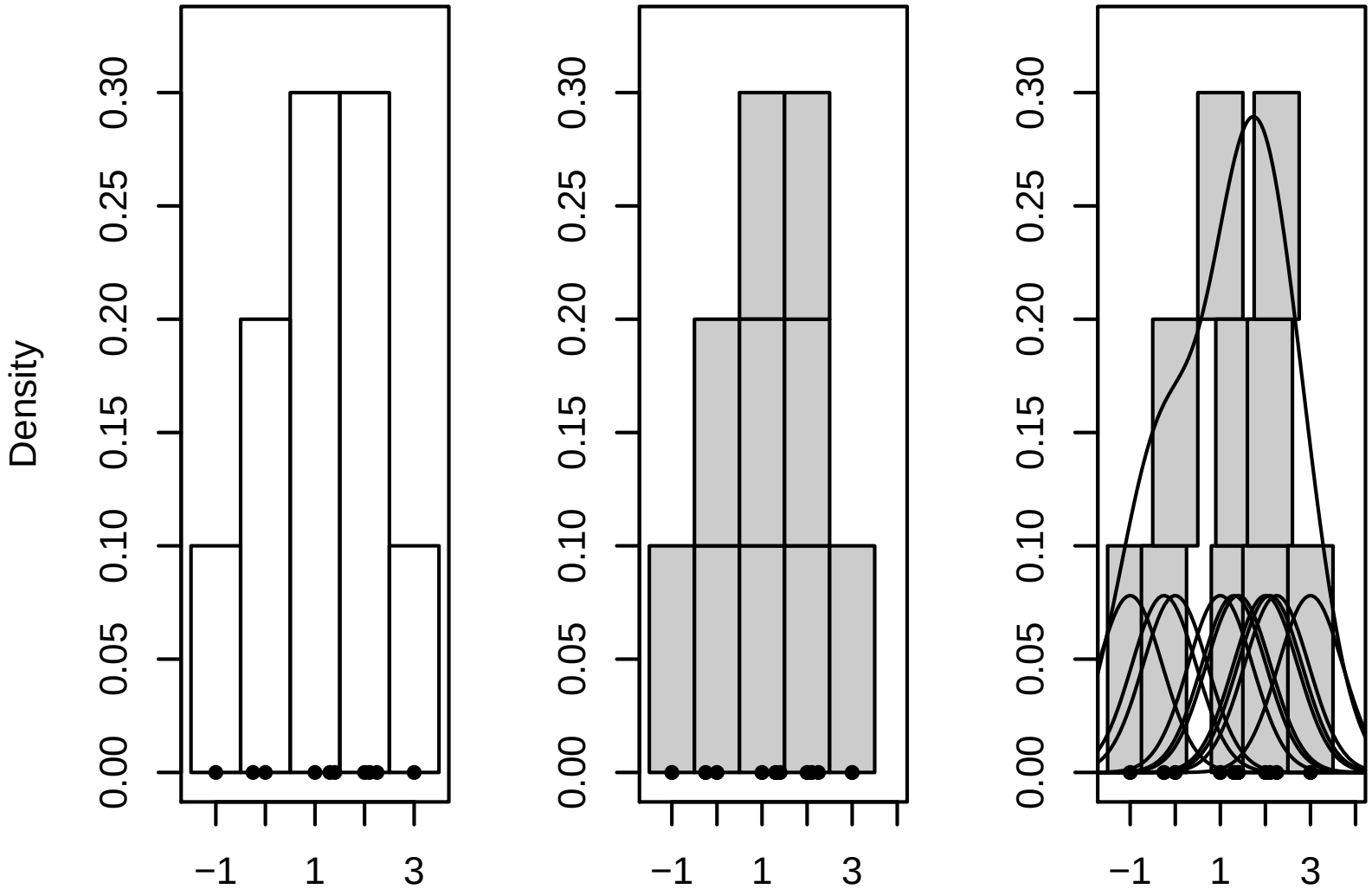
How to Estimate the Density?

- Given data, we wish to estimate the density.
- The most familiar way is via the histogram.
- Another, very powerful tool is the kernel estimator, which can be thought of as a kind of "variable bin histogram".
- This is easiest to understand visually.

The Histogram



The Histogram – The Kernel Estimator



The Kernel Estimator

For n observations $\{x_1, \ldots, x_n\}$, we define

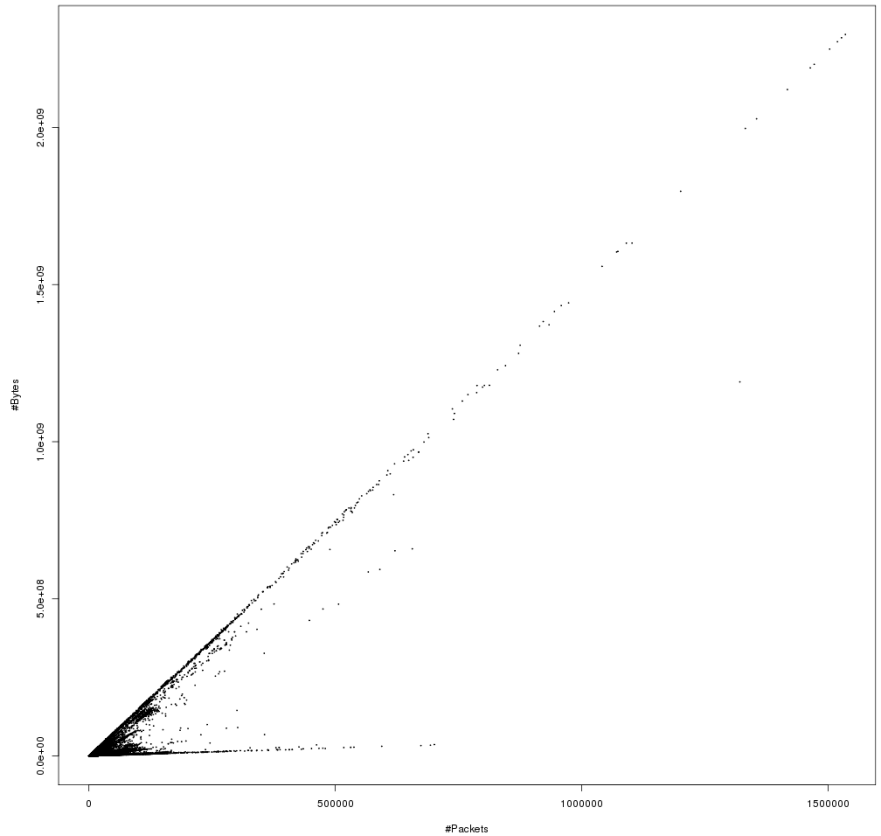
$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) = \frac{1}{nh} \sum_{i=1}^{n} \phi\left(\frac{x - x_i}{h}\right) = \frac{1}{n} \sum_{i=1}^{n} \phi(x; x_i, h).$$

- Here K is the *kernel* – a unimodal density. ϕ is the normal (Gaussian) density.
- Easily extended to multivariate versions.
- Note that this is an "average".

Poor Man's Kernel Estimator

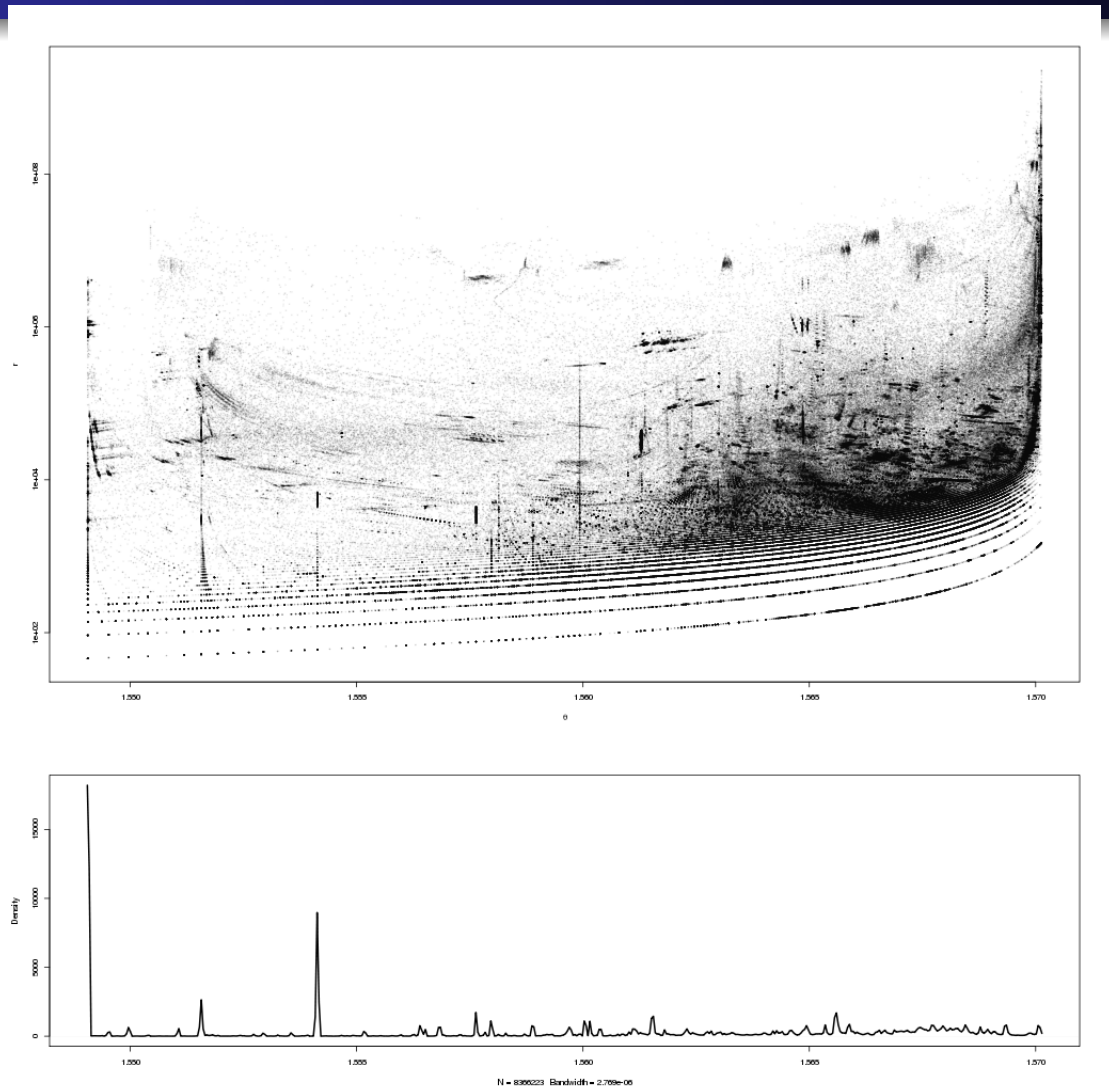
- In 2-d, one can plot the density using surface plots or contour plots.
- Alternatively, one can plot it as an image, with gray-scale indicating the density.
- Plotting a scatterplot using alpha blending is essentially a poor-man's version of this (with "round" kernels).

Network Flows



http://csr.lanl.gov/data/cyber1/

Network Flows



Density Estimation for Classification

- The class conditional probability densities provide a way of performing classification:
  - Obtain (estimate) the probability density function for each class.
  - For each new observation, compute it's value for each density – this is called the *likelihood* – and classify according to the largest.

Densities and Estimation

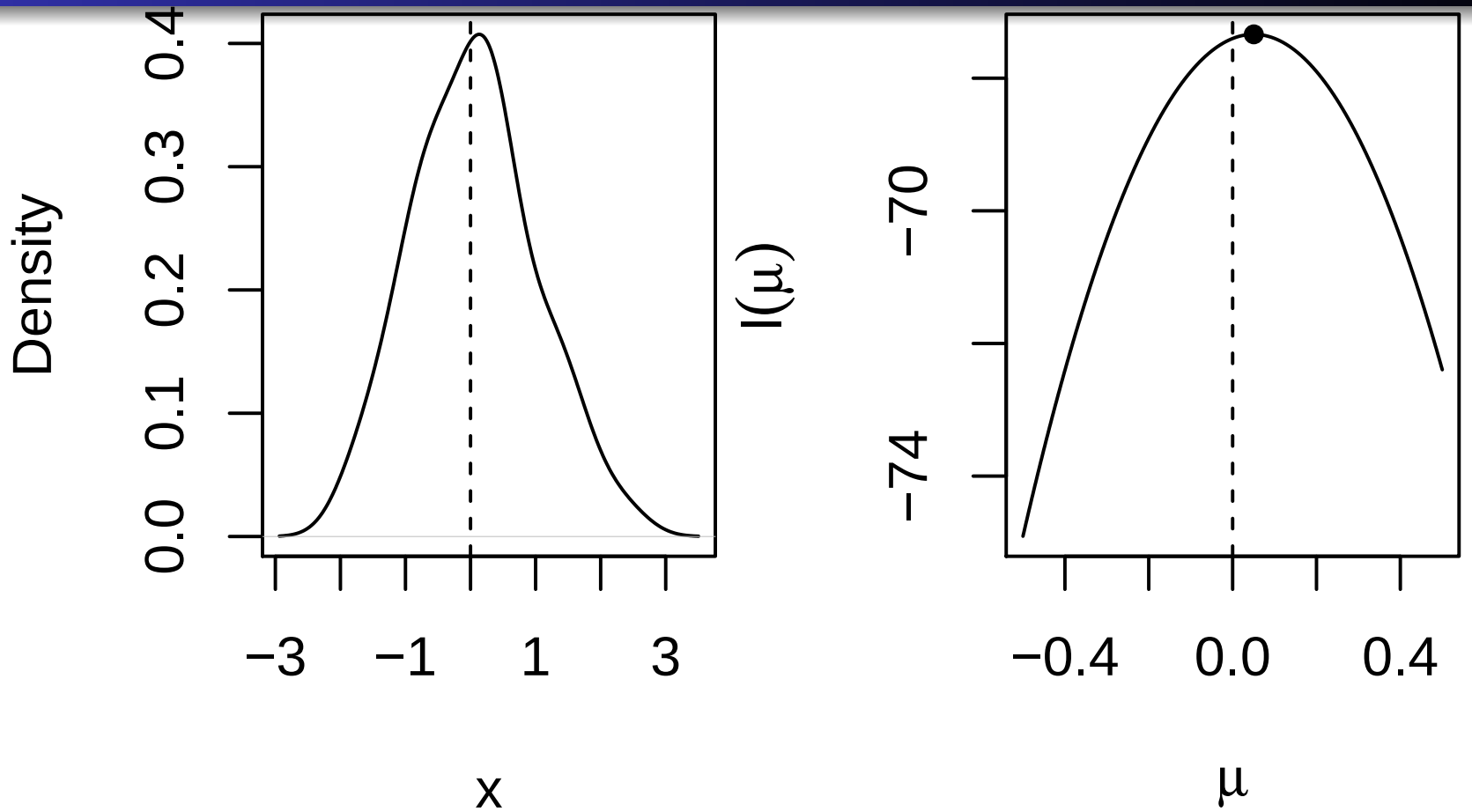
- The value of the density – the likelihood – can be used to fit the parameters of the model. For n observations $\{x_1, \ldots, x_n\}$:

$$L(\theta) = \prod_{i=1}^{n} f(x_i; \theta),$$

$$\ell(\theta) = \sum_{i=1}^{n} \log(f(x_i; \theta)).$$

- Choose θ to maximize $\ell(\theta)$ – maximum likelihood.

Illustration



In this case the maximum likelihood estimator for μ is the sample mean.

Streaming Data

Averages can be computed in a streaming fashion:

$$\widehat{X}_n = \frac{n-1}{n}\widehat{X}_{n-1} + \frac{1}{n}X_n.$$

We can implement an exponential window:

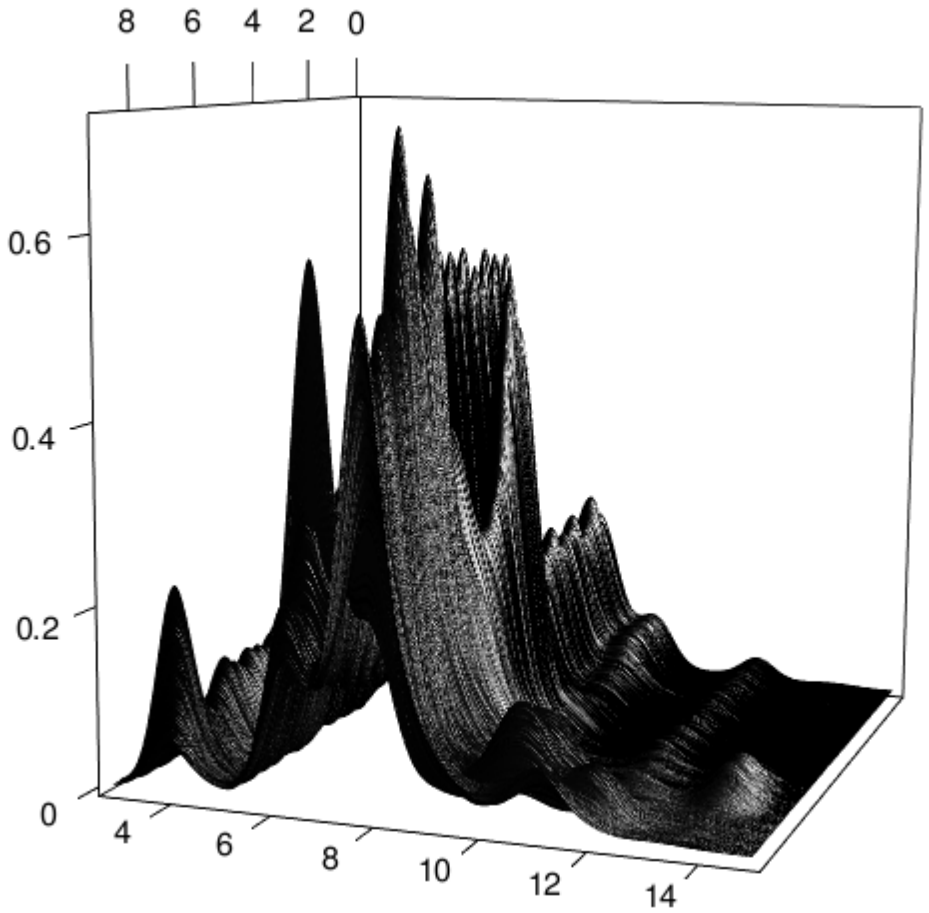
$$\widetilde{X}_n = \frac{N-1}{N}\widetilde{X}_{n-1} + \frac{1}{N}X_n = \theta\widetilde{X}_{n-1} + (1-\theta)X_n,$$

and apply this idea to the kernel estimator:

$$\widetilde{f}_n(x) = \theta\widetilde{f}_{n-1}(x) + (1-\theta)\phi\left(\frac{x - X_n}{h}\right).$$

θ controls how much of the past we "remember". Note that we have to set a grid of x points at which we want to compute $\widetilde{f}$.

Streaming Network Flows: log(#bytes) in a Flow



Topics

Regression

- Regression is the name for fitting a model of the form:

$$y = f(x) + \epsilon.$$

- There are more general versions of this, but the idea is that we observe a noisy version of (x, y) pairs, and want to find f.
- The simplest such model is linear regression:

$$\begin{aligned} y &= x\beta_1 + \beta_0 + \epsilon \\ &= x'\beta + \epsilon \end{aligned}$$

where we have put β_0 into β and augmented x with a column of 1s.

Linear Regression

Under least squares, it is possible (with linear algebra) to find an exact formula for β. Least squares is:

$$\sum (y_i - \widehat{y}_i)^2 = \|y - \widehat{y}\|_2^2.$$

Then

$$\widehat{\beta} = \arg\min_{\beta} \|y - \widehat{y}\|_2,$$

and one can find β in closed form.

Ridge Regression

Ridge regression adds a penalty, which forces the algorithm to try to choose "smaller" features:

$$\begin{aligned}\widehat{\beta} &= \arg\min_{\beta} \|y - \widehat{y}\|_2 + \lambda \sum |\beta|^2 \\ &= \arg\min_{\beta} \|y - \widehat{y}\|_2 + \lambda \|\beta\|_2^2.\end{aligned}$$

Lasso Regression

The Lasso adds a penalty, which forces the algorithm to choose a sparse representation – set many of the parameters to 0.

$$\widehat{\beta} = \arg\min_{\beta} \|y - \widehat{y}\|_2 + \lambda \sum |\beta|.$$

Putting These Together: Elastic Net Regularization

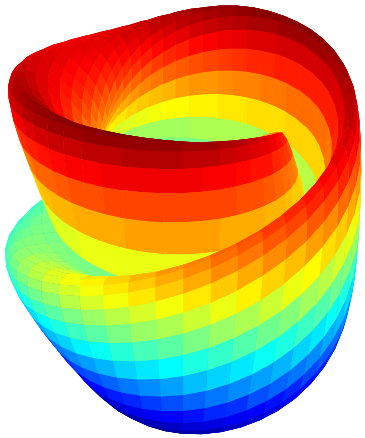
$$\widehat{\beta} = \arg\min_{\beta} \|y - \widehat{y}\|_2 + \lambda_2 \|\beta\|_2^2 + \lambda_1 \|\beta\|_1.$$

This trades off the two penalties. The ℓ_1 ensures a sparse model, while the ℓ_2 term tends to stabilize the sparsity – keeps it from being "too sparse" – and helps with "grouped variables" – highly correlated variables.

Topics

Manifold Learning

- Hypothesis: high dimensional data "lives" on a lower dimensional structure.
- Manifold learning is a set of techniques to infer this structure, or to embed the data from the high dimensional space into a lower dimensional space that respects the local structure.



Multidimensional Scaling

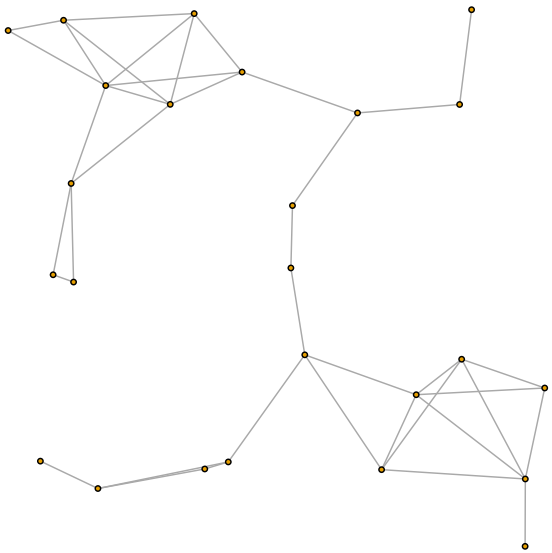
- Problem: Given a distance matrix (or dissimilarity matrix) D, find a set of points $X \in \mathbb{R}^d$ whose distance $d(X)$ best approximates D.
- This is the problem solved by multidimensional scaling (MDS).
- Different definitions of "best approximates" lead to different algorithms.
- Classical MDS utilizes the eigenvector decomposition of (a modified version of) the distance matrix.
- Some manifold learning algorithms compute a local distance and use MDS, others compute eigenvectors of related matrices. These are the algorithms I use most often.

Basic Graph Theory

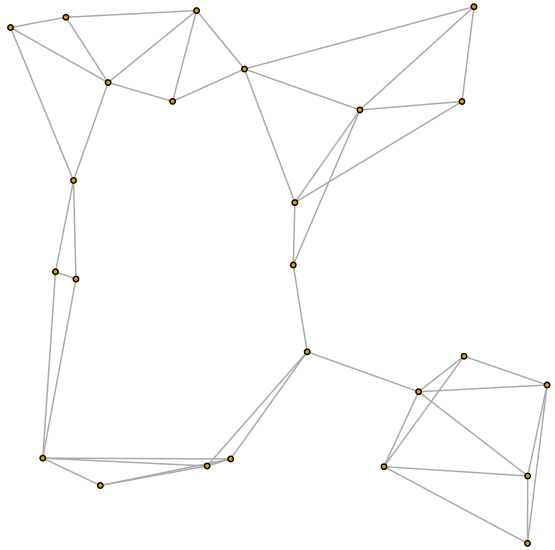
- A graph is a set V of vertices, and E of pairs of vertices (edges).
- The edges can be directed or undirected, and can have weights.
- In this talk they will be undirected.
- The (graph) distance between two vertices is the length of the shortest path between them in the graph.
- The adjacency matrix of a graph on n vertices is the $n \times n$ binary matrix with a 1 in those positions corresponding to the edges of the graph.
- The spectrum of a graph is the eigen decomposition of the adjacency matrix A, or more generally, of some function $f(A)$.

Graph Examples

ϵ-ball graph with $\epsilon = 0.25$.



3-nearest neighbor graph.



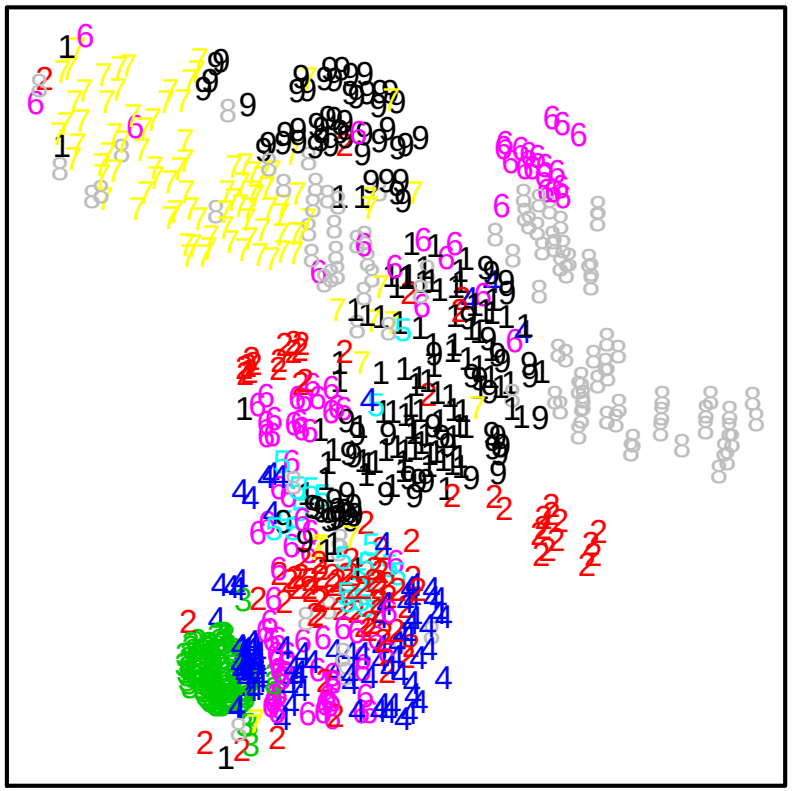
Basic Steps of Manifold Learning

- Given data $\{x_1, \ldots, x_n\} \in \mathbb{R}^p$:
  1. Construct a graph whose vertices are the x_i with edges between "near" points.
     - k-nearest neighbor graph.
     - ϵ-ball graph.
     - Variations.
  2. Compute the eigenvectors of:
     - The adjacency matrix.
     - The Laplacian of the adjacency matrix.
     - Scaled or modified versions of the above.
  3. Set Z to the matrix with columns corresponding to the main eigenvectors. That is, the rows $\{z_1, \ldots, z_n\}$ are the "embedded" data.
- Perform inference on Z.

Manifold Learning

Compute an ϵ-ball graph on a subset of the Kaggle malware data.

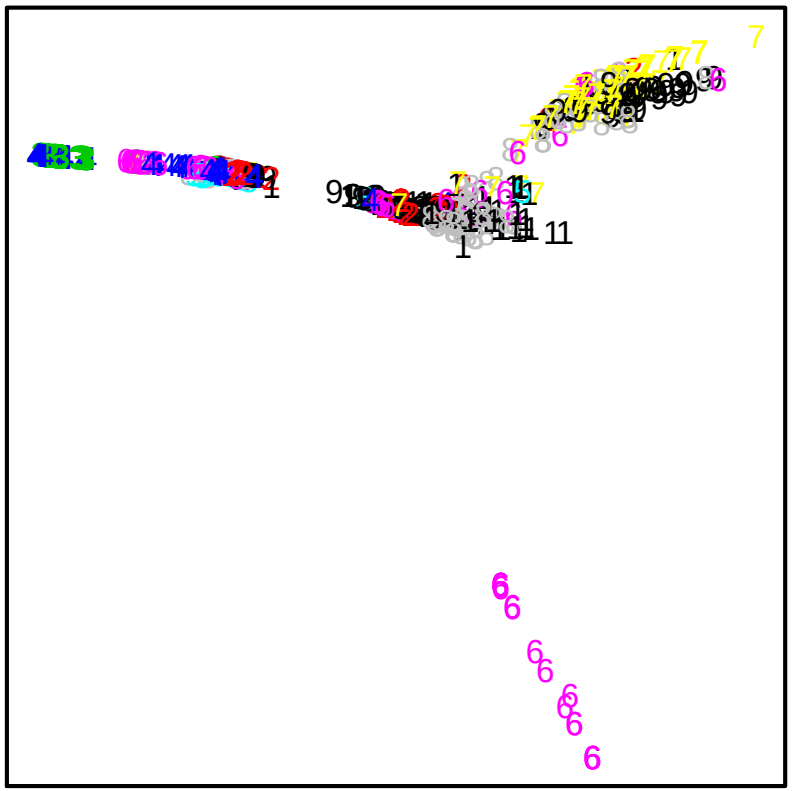
- **Layout the graph.**
- Embed using scaled Laplacian.
- Embed using adjacency matrix.
- Embed using MDS on graph distance.



Manifold Learning

Compute an ϵ-ball graph on a subset of the Kaggle malware data.

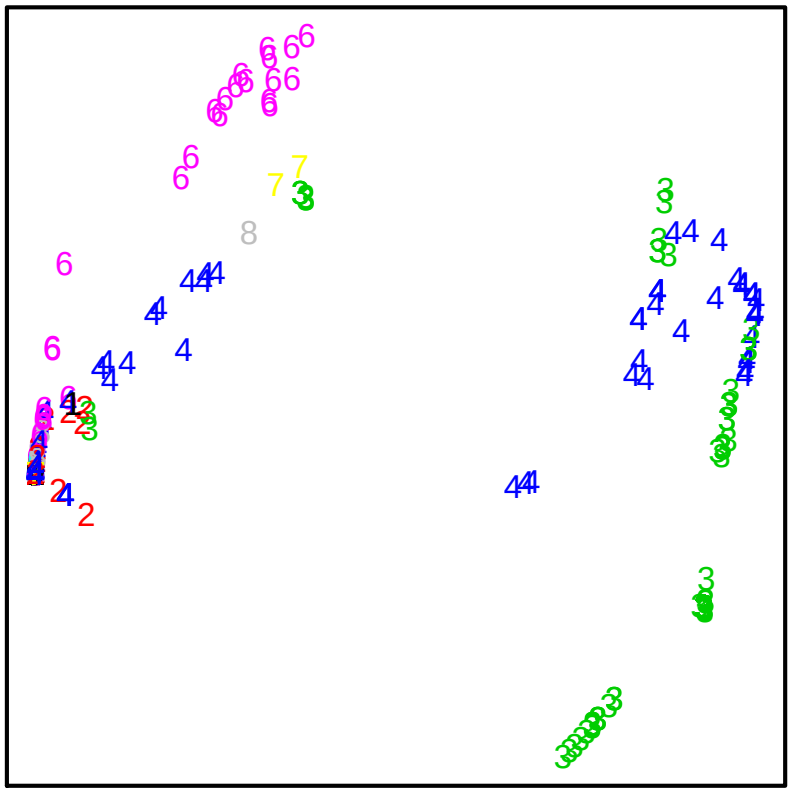
- Layout the graph.
- **Embed using scaled Laplacian.**
- Embed using adjacency matrix.
- Embed using MDS on graph distance.



Manifold Learning

Compute an ϵ-ball graph on a subset of the Kaggle malware data.

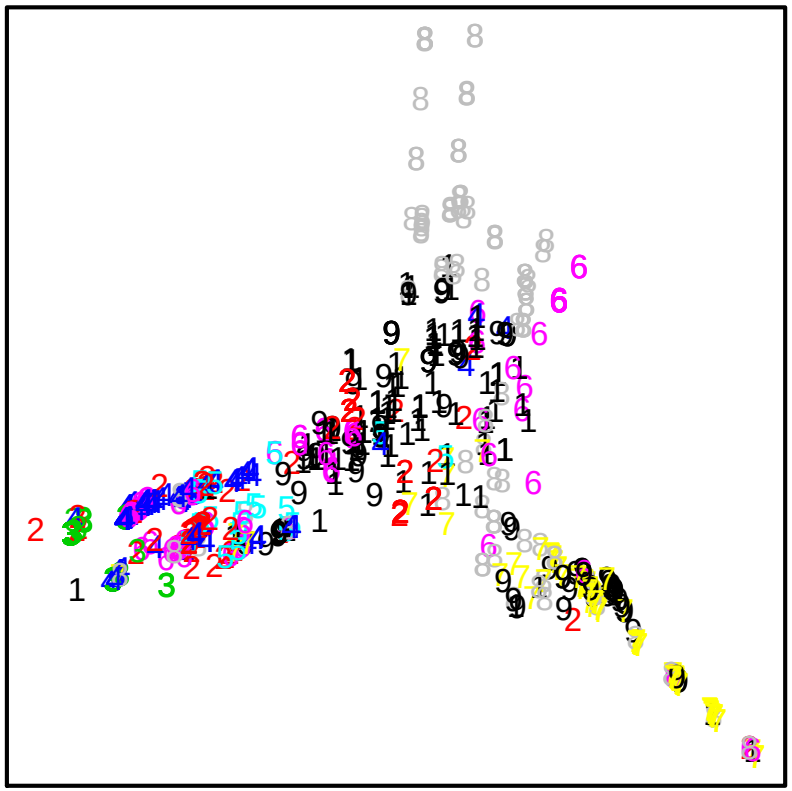
- Layout the graph.
- Embed using scaled Laplacian.
- **Embed using adjacency matrix.**
- Embed using MDS on graph distance.



Manifold Learning

Compute an ϵ-ball graph on a subset of the Kaggle malware data.

- Layout the graph.
- Embed using scaled Laplacian.
- Embed using adjacency matrix.
- **Embed using MDS on graph distance.**



Manifold Learning Discussion

- Different embedding methods extract different information about the data.
- These two dimensional plots are misleading in that there is no reason to assume the intrinsic dimensionality is 2.
- Some care must be taken to ensure that the embedding method can be applied to new data.

Discussion

- Mathematics has many tools for the data analyst, in particular for the analysis of cyber data.
- These tools include:
  - Probability
  - Statistics
  - Computational statistics.
  - Machine learning.
  - Graph theory.
  - Manifold learning.
  - Topological data analysis.
- New applications of "pure" mathematics to data analysis are developed every day, and these areas are all huge growth areas for applied mathematicians.