A Few Interesting Connections Between Statistics and Computer Science

Palash Sarkar

Applied Statistics Unit
Indian Statistical Institute, Kolkata
India
palash@isical.ac.in

Indian Statistical Institute, North-East Centre, Tezpur
24th July 2011
“Statistics is the universal tool of inductive inference, research in natural and social sciences, and technological applications.

Statistics, therefore, must always have purpose, either in the pursuit of knowledge or in the promotion of human welfare.”

– Prasanta Chandra Mahalanobis
(2nd December, 1956)
“It is unworthy of excellent men to lose hours like slaves in the labour of calculation which could be relegated to anyone else if machines were used.”

– Gottfried von Leibniz
“It is unworthy of excellent men to lose hours like slaves in the labour of calculation which could be relegated to anyone else if machines were used.”

– Gottfried von Leibniz

“A huge gap exists between what we know is possible with today’s machines and what we have so far been able to finish.”

– Donald Knuth
Charles Babbage (1792–1871)

- Difference Engine and Analytical Engine.
  - Precursor of modern computers.
  - Architecture similar in spirit to modern computer architecture.
  - The analytical engine was the design of a programmable computer.
Charles Babbage (1792–1871)

- **Difference Engine and Analytical Engine.**
  - Percursor of modern computers.
  - Architecture similar in spirit to modern computer architecture.
  - The analytical engine was the design of a programmable computer.
- **Cryptanalysis of the Vigenére cipher (le chiffre indéchiffrable).**
  - Statistical cryptanalysis: introduced the idea of correlation.
  - Required algorithmic ideas to carry out the computation.
  - Used by the British in the Crimean war (Simon Singh).
Charles Babbage (1792–1871)

- Difference Engine and Analytical Engine.
  - Percursor of modern computers.
  - Architecture similar in spirit to modern computer architecture.
  - The analytical engine was the design of a programmable computer.

- Cryptanalysis of the Vigenére cipher (le chiffre indéchiffrable).
  - Statistical cryptanalysis: introduced the idea of correlation.
  - Required algorithmic ideas to carry out the computation.
  - Used by the British in the Crimean war (Simon Singh).

- Mathematical tables.
  - Mortality tables used by the life insurance industry.
Charles Babbage (1792–1871)

- Difference Engine and Analytical Engine.
  - Percursor of modern computers.
  - Architecture similar in spirit to modern computer architecture.
  - The analytical engine was the design of a programmable computer.
- Cryptanalysis of the Vigenére cipher (le chiffre indéchiffrable).
  - Statistical cryptanalysis: introduced the idea of correlation.
  - Required algorithmic ideas to carry out the computation.
  - Used by the British in the Crimean war (Simon Singh).
- Mathematical tables.
  - Mortality tables used by the life insurance industry.

Tennyson wrote:
“Every moment dies a man, Every moment one is born.”
Charles Babbage (1792–1871)

- Difference Engine and Analytical Engine.
  - Percursor of modern computers.
  - Architecture similar in spirit to modern computer architecture.
  - The analytical engine was the design of a programmable computer.
- Cryptanalysis of the Vigenère cipher (le chiffre indéchiffrable).
  - Statistical cryptanalysis: introduced the idea of correlation.
  - Required algorithmic ideas to carry out the computation.
  - Used by the British in the Crimean war (Simon Singh).
- Mathematical tables.
  - Mortality tables used by the life insurance industry.

Tennyson wrote:
“Every moment dies a man, Every moment one is born.”

Babbage wrote back:
“Every moment dies a man, Every moment 1 1/16 is born.”
Charles Babbage (1792–1871)

- Difference Engine and Analytical Engine.
  - Percursor of modern computers.
  - Architecture similar in spirit to modern computer architecture.
  - The analytical engine was the design of a programmable computer.

- Cryptanalysis of the Vigenère cipher (le chiffre indéchiffrable).
  - Statistical cryptanalysis: introduced the idea of correlation.
  - Required algorithmic ideas to carry out the computation.
  - Used by the British in the Crimean war (Simon Singh).

- Mathematical tables.
  - Mortality tables used by the life insurance industry.

_Tennyson wrote:_
“Every moment dies a man, Every moment one is born.”

_Babbage wrote back:_
“Every moment dies a man, Every moment 1 1/16 is born.”

_With the comment:_ “1 1/16 will be sufficiently accurate for poetry.”
- Jute survey and the travelling salesman problem (TSP).
- Order statistics and clustering.
- Design of experiments, coding theory and computer science.
- A sampling problem.
Jute Survey and the Travelling Salesman Problem.
Jute Survey

Jute Survey


- A very important work in the area of large-scale sample surveys.
  - The preparatory work itself required handling of 55,000 *mauza* sheets, locating 42,000 random points and listing 8,96,938 individual plots.

- Introduced the idea of pilot surveys to collect information on the cost and variance function necessary for planning a large scale survey.
Jute Survey


- A very important work in the area of large-scale sample surveys.
  - The preparatory work itself required handling of 55,000 *mauza* sheets, locating 42,000 random points and listing 8,96,938 individual plots.

- Introduced the idea of pilot surveys to collect information on the cost and variance function necessary for planning a large scale survey.

- Early demonstration of “the possibility of using large-scale sample surveys to collect information quickly and economically with sufficient accuracy.” (C. R. Rao in the introduction to the version of the paper printed in Sankhyā, Vol. 29, 1967).

Palash Sarkar (ISI, Kolkata)
The Travelling Salesman Problem

Input: $n$ cities and the pairwise cost of travelling from one city to another.

Requirement: a salesman has to visit each city exactly once and wants to do so with the minimum possible cost.
The Travelling Salesman Problem

**Input:** \( n \) cities and the pairwise cost of travelling from one city to another.

**Requirement:** a salesman has to visit each city exactly once and wants to do so with the minimum possible cost.

- Known to be NP-complete (i.e., complete for the class of problems which can be solved in non-deterministic polynomial time).
- Finding provably good approximations is also computationally hard.
- Arguably the most famous and extensively studied of all NP-complete problems.
Suppose that the area has been divided into $k$ zones and $A_i$ is the area in the $i$-th zone.

$$v_i = \psi(p_i, x_i) \text{ and } t_i = \phi(x_i, y_i), 1 \leq i \leq k.$$ 

- $p_i$ is the proportion of land under jute in the $i$-th zone,
- $x_i$ is the size of grids in the $i$-th zone,
- $y_i$ is the density of number of grids per square mile in the $i$-th zone.
Jute Survey: Variance and Cost Functions

Suppose that the area has been divided into \( k \) zones and \( A_i \) is the area in the \( i \)-th zone.

\[
v_i = \psi(p_i, x_i) \quad \text{and} \quad t_i = \phi(x_i, y_i), \quad 1 \leq i \leq k.
\]

- \( p_i \) is the proportion of land under jute in the \( i \)-th zone,
- \( x_i \) is the size of grids in the \( i \)-th zone,
- \( y_i \) is the density of number of grids per square mile in the \( i \)-th zone.

Variance and total cost are given by

\[
V = \sum_{i=1}^{k} \frac{A_i v_i}{y_i} \quad T = \sum_{i=1}^{k} A_i t_i.
\]
Suppose that the area has been divided into \( k \) zones and \( A_i \) is the area in the \( i \)-th zone.

\[
v_i = \psi(p_i, x_i) \quad \text{and} \quad t_i = \phi(x_i, y_i), \quad 1 \leq i \leq k.
\]

- \( p_i \) is the proportion of land under jute in the \( i \)-th zone,
- \( x_i \) is the size of grids in the \( i \)-th zone,
- \( y_i \) is the density of number of grids per square mile in the \( i \)-th zone.

Variance and total cost are given by

\[
V = \sum_{i=1}^{k} A_i v_i / y_i \quad T = \sum_{i=1}^{k} A_i t_i.
\]

The problem that Mahalnobis considers is to fix the cost to a certain value and then choose \((x_i, y_i)\) such that \( V \) is minimised.
Analysis of the Cost Function

- The actual cost in Rupees is determined by time.
- The major component of the total time is the time to move from one plot to another.
- This is the key to the connection to TSP!
The actual cost in Rupees is determined by time.
The major component of the total time is the time to move from one plot to another.
This is the key to the connection to TSP!

Assume $n$ sampling units (abstracted as points) are scattered in a given area.
Assume that movement from one point to another can be done in a straight line with cost proportional to the Euclidean distance between the points.
The expected length of a TSP tour is $\sqrt{n} - 1/\sqrt{n}$.
  \begin{itemize}
    \item Mahalanobis remarks that this is easy to see.
  \end{itemize}
Length of Random TSP Instances

**Beardwood-Halton-Hammersley Theorem (1959).**

If \(x_1, \ldots, x_n\) are uniformly distributed on the unit square and \(T_n\) is the length of a TSP which connects these points, then with probability one

\[
\lim_{n \to \infty} \frac{T_n}{\sqrt{n}} \to \beta
\]

as \(n\) tends to infinity. Here \(\beta\) is a constant.
Length of Random TSP Instances

If $x_1, \ldots, x_n$ are uniformly distributed on the unit square and $T_n$ is the length of a TSP which connects these points, then with probability one

$$\lim_{n \to \infty} \frac{T_n}{\sqrt{n}} \to \beta$$

as $n$ tends to infinity. Here $\beta$ is a constant.

- The above is actually a special case of the BHH theorem which holds over dimensions $\geq 2$.
- Later works have used martingale arguments to show that the TSP length (which is a random variable) is concentrated tightly about its mean.
- A recent result of this type is due to Kannan (2009) where such a result is shown as an application of a sharper version of the Höffding-Azuma inequality for martingales.
Order statistics and clustering.
Suppose there is a list of \( n \) (unsorted) values and we wish to find the median of these values.
Finding Median

Suppose there is a list of \( n \) (unsorted) values and we wish to find the median of these values.

- **Strategy**: sort the elements and pick the middle one.
  - Sorting: a classical problem of computer science;
  - well understood; several well known algorithms;
  - (comparison) based sorting requires time \( \theta(n \log n) \).
Suppose there is a list of \( n \) (unsorted) values and we wish to find the median of these values.

- **Strategy:** sort the elements and pick the middle one.
  - Sorting: a classical problem of computer science;
  - well understood; several well known algorithms;
  - (comparison) based sorting requires time \( \theta(n \log n) \).

Can we do better?
Problem: Find the $k$-th smallest element in a list of $n$ unsorted elements.

- Strategy: finding repeated minimum; requires time $O(kn)$. 
Problem: Find the $k$-th smallest element in a list of $n$ unsorted elements.

- Strategy: finding repeated minimum; requires time $O(kn)$.
- Strategy: choose a pivot and in linear time partition the list into two parts with the pivot in its proper position; we need to recursively process only one of these two parts.
Problem: Find the $k$-th smallest element in a list of $n$ unsorted elements.

- Strategy: finding repeated minimum; requires time $O(kn)$.
- Strategy: choose a pivot and in linear time partition the list into two parts with the pivot in its proper position; we need to recursively process only one of these two parts.
  - Performs well in practice; expected $O(n)$ comparisons.
  - In-place algorithm.
  - Sensitive to the choice of the pivot; $O(n^2)$ comparisons in the worst case; choosing random pivots works well in practice.
Problem: Find the $k$-th smallest element in a list of $n$ unsorted elements.

- **Strategy “median of medians”:**
  - **goal:** always find a ‘good’ median;
  - divide the list into groups of 5 elements;
  - find the median for each group of 5 elements;
  - recursively find the median of the sublist of $n/5$ elements;
  - this becomes the pivot.
Problem: Find the $k$-th smallest element in a list of $n$ unsorted elements.

- Strategy “median of medians”:
  - goal: always find a ‘good’ median;
  - divide the list into groups of 5 elements;
  - find the median for each group of 5 elements;
  - recursively find the median of the sublist of $n/5$ elements;
  - this becomes the pivot.
  - Worst case running time is $O(n)$.
  - In practice the partitioning strategy with random pivots works better.
Problem: Find the $k$-th smallest element in a list of $n$ unsorted elements.

- **Strategy “median of medians”:**
  - goal: always find a ‘good’ median;
  - divide the list into groups of 5 elements;
  - find the median for each group of 5 elements;
  - recursively find the median of the sublist of $n/5$ elements;
  - this becomes the pivot.
    - Worst case running time is $O(n)$.
    - In practice the partitioning strategy with random pivots works better.

Problem: given a list of data items, partition it into “similar” groups.

- A natural question when there is a list of data items.
- Applications to data mining, information retrieval, image processing and web search.
Clustering

**Problem:** given a list of data items, partition it into “similar” groups.

- A natural question when there is a list of data items.
- Applications to data mining, information retrieval, image processing and web search.

**k-means clustering:** a widely used definition. Given a set of points $P$, the k-means clustering problems seeks to find a set $K$ of $k$ centers, such that

$$\sum_{p \in P} d(p, K)^2$$

is minimised.
**k-Means Clustering**

- NP-hard even for $k = 2$.
- Best known algorithm for the *exact* $k$-means problem has complexity $\Omega(n^d)$ for $n$ points in $d$ dimensions.
- Approximation algorithms: for constant $\varepsilon$ and $k$, there is a $(1 + \varepsilon)$-algorithm for the $k$-means problem running in time $O(nd)$ (Kumar-Sabharwal-Sen, 2004).
**k-Means Clustering**

- NP-hard even for $k = 2$.
- Best known algorithm for the exact $k$-means problem has complexity $\Omega(n^d)$ for $n$ points in $d$ dimensions.
- Approximation algorithms: for constant $\varepsilon$ and $k$, there is a $(1 + \varepsilon)$-algorithm for the $k$-means problem running in time $O(nd)$ (Kumar-Sabharwal-Sen, 2004).

**Basic idea.**

- The centroid of a set of points can be very well approximated by sampling a constant number of points and finding the centroid of the sample (Inaba-Katoh-Imai, 1994).
- Sampling $O(k)$ points and considering all constant size subsets of the sample can give the centres of the largest clusters.
- A more careful strategy is required to find the centres of the smaller clusters.
**Problem:** Data points are generated by a mixture of $k$ probability distributions.
**Problem:** Data points are generated by a mixture of $k$ probability distributions.

- If the means of every pair of distributions are $\text{poly}(k)$ times standard deviations apart, then the mixture can be learnt in polynomial time.
- Uses properties of the generative models and properties of particular distributions.
Theorem (Kumar-Kannan, 2010): If the projection of any data point onto the line joining its cluster center to any other cluster center is $\Omega(k)$ times standard deviations closer to its own center than the other center (a proximity condition) then clustering can be done correctly in polynomial time.
**Theorem (Kumar-Kannan, 2010):** If the projection of any data point onto the line joining its cluster center to any other cluster center is $\Omega(k)$ times standard deviations closer to its own center than the other center (a proximity condition) then clustering can be done correctly in polynomial time.

- The role of standard deviation is played by (a scaled version of) the spectral norm of the $A - C$ where $A$ is the matrix representing the set of points and $C$ is the matrix representing the corresponding cluster centres.

- Based on the $k$-means algorithm: shows that the $k$-means algorithm converges to the true centres provided the initial estimates of the centres are close enough to the true centres and all but an $\varepsilon$ fraction of the points satisfy the proximity condition.
Design of experiments, coding theory and computer science.
Ronald Fisher

- Founder of the modern methods of design and analysis of experiments.
  - Experimental methods existed previously.
  - But, there was no coherent theory on the estimation of errors from the results themselves.
Ronald Fisher

- Founder of the modern methods of design and analysis of experiments.
  - Experimental methods existed previously.
  - But, there was no coherent theory on the estimation of errors from the results themselves.
- In 1935 wrote a book entitled “The Design of Experiments”.
  - Detailed discussion of the basic logical principles of experimentation.
Ronald Fisher

- Founder of the modern methods of design and analysis of experiments.
  - Experimental methods existed previously.
  - But, there was no coherent theory on the estimation of errors from the results themselves.
- In 1935 wrote a book entitled “The Design of Experiments”.
  - Detailed discussion of the basic logical principles of experimentation.
- Combinatorial aspect of designs.
  - Enumeration of $5 \times 5$ and $6 \times 6$ latin squares.
  - Investigated the existence of balanced incomplete block designs (BIBDs).
  - Showed relationship between BIBD and complete orthogonal sets of latin squares.
Construction of several infinite families of BIBDs (1939).
- Use of finite (Galois) fields.
- Introduction of the difference method.
- Idea of recursive construction, i.e., building larger designs from smaller ones (jointly with Srikhande and Hanani).
Raj Chandra Bose

- Construction of several infinite families of BIBDs (1939).
  - Use of finite (Galois) fields.
  - Introduction of the difference method.
  - Idea of recursive construction, i.e., building larger designs from smaller ones (jointly with Srikhande and Hanani).

- BCH codes (1960): a general family of codes (joint work with D. K. Ray-Chaudhuri and independently obtained by Hocquenghem (1959)).
  - Based on the algebra of finite fields.
  - The number of errors that can be corrected can be pre-specified.
  - Important classes of codes (such as Reed-Solomon codes) can be seen as special cases of BCH codes.
  - Later independent work by Berlekamp and Massey provided simple methods for decoding BCH codes.
Combinatorial designs and error-correcting codes are related.

Example: The matrix listing the codewords of an ECC forms an orthogonal array.
Combinatorial designs and error-correcting codes are related.
- Example: The matrix listing the codewords of an ECC forms an orthogonal array.

Pseudo-random generator: trading hardness for randomness.
- A fundamental result due to Nisan and Wigderson.
- The core of the construction is the following object: A collection of sets \( \{S_1, \ldots, S_n\} \), where \( S_i \subset \{1, \ldots, l\} \) satisfying:
  \[ |S_i| = m \] and for \( i \neq j \), \( |S_i \cap S_j| \leq k \).
Combinatorial designs and error-correcting codes are related.

- Example: The matrix listing the codewords of an ECC forms an orthogonal array.

Pseudo-random generator: trading hardness for randomness.

- A fundamental result due to Nisan and Wigderson.
- The core of the construction is the following object: A collection of sets \( \{S_1, \ldots, S_n\} \), where \( S_i \subset \{1, \ldots, l\} \) satisfying:
  \[ |S_i| = m \text{ and for } i \neq j, |S_i \cap S_j| \leq k. \]

(Almost universal) hash function: \( \{H_\tau\} \) such that for uniform random \( \tau \) and distinct \( M \) and \( M' \), \( \Pr[H_\tau(M) = H_\tau(M')] \leq \varepsilon. \)

- Extremely important in cryptography and computer science.
- Closely related to codes and orthogonal arrays.
A sampling problem
Consider an internet router.

A huge number of internet packets are flowing through the router.

Only a sample of the packets can be stored.

- Reservoir sampling, Knuth 1969.

Each packet has a weight which is the number of bytes in the packet.

Analysis based on the sample.

- Estimate weights of arbitrary subsets of the packets that have gone by.
- The subsets whose weights are to be analysed are not known at the time of sampling.
All sales data are stored.
Later analysis.
  A sample is checked against weather records to estimate the number of days of rain before a boom in rain-gear sales.
  The questions for the analysis may not be known at the time of sampling.
Can be related to reservoir sampling.
  A small reservoir of samples can be easily shared over the internet by analysts at different locations.
Priority Sampling


Set-up and requirements.

- Each data item has a weight.
- Estimation from the samples must be accurate even with heavy-tailed distributions where most of the weight is concentrated on a few items.
- Sample should be weight sensitive, giving priority to heavy items.
- Sampling without replacement so that heavy items are not selected multiple times.
Priority Sampling Technique

- Items are $i = 0, 1, \ldots, n$ with weights $w_i$.
- For item $i$, a uniform random number $\alpha_i \in (0, 1]$ is generated and item $i$ is given a priority $q_i = w_i / \alpha_i$. 
Priority Sampling Technique

- Items are $i = 0, 1, \ldots, n$ with weights $w_i$.
- For item $i$, a uniform random number $\alpha_i \in (0, 1]$ is generated and item $i$ is given a priority $q_i = w_i / \alpha_i$.
- The sample $S$ consists of the $k$ highest priority items.
- Let $\tau$ be the $(k + 1)$st priority.
- Each sampled item $i$ in $S$ gets a weight estimate $\hat{w}_i = \max\{w_i, \tau\}$, while non-sampled items get weight estimate $\hat{w}_i = 0$. 

Palash Sarkar  (ISI, Kolkata)  Stat-CS Connections  Tezpur, 2011  28 / 31
Priority Sampling Technique

- Items are $i = 0, 1, \ldots, n$ with weights $w_i$.
- For item $i$, a uniform random number $\alpha_i \in (0, 1]$ is generated and item $i$ is given a priority $q_i = w_i / \alpha_i$.
- The sample $S$ consists of the $k$ highest priority items.
- Let $\tau$ be the $(k + 1)$st priority.
- Each sampled item $i$ in $S$ gets a weight estimate $\hat{w}_i = \max\{w_i, \tau\}$, while non-sampled items get weight estimate $\hat{w}_i = 0$.
- Reservoir sampling: easy to implement priority sampling in the reservoir context by using a priority queue data structure.
  - Point to note: the feasibility of the sampling strategy is inherently linked to a suitable and efficient data structure.
Unbiased estimate: $E[\hat{w}_i] = w_i$.

By linearity of expectation, this extends to unbiased estimates of arbitrary subset sums.

The variance of the estimates can be estimated.

Covariance between estimates $\hat{w}_i$ and $\hat{w}_j$ of different weights is zero.
Unbiased estimate: $E[\hat{w}_i] = w_i$.

By linearity of expectation, this extends to unbiased estimates of arbitrary subset sums.

The variance of the estimates can be estimated.

Covariance between estimates $\hat{w}_i$ and $\hat{w}_j$ of different weights is zero.

A strong near-optimality condition: For any weight sequence, there exists no specialised scheme for sampling $k$ items with unbiased weight estimators that gets smaller variance sum than priority sampling with $(k + 1)$ items.

(Conjectured by DLT and settled by Szegedy (2006).)
Unbiased estimate: \( E[\hat{w}_i] = w_i \).

By linearity of expectation, this extends to unbiased estimates of arbitrary subset sums.

The variance of the estimates can be estimated.

Covariance between estimates \( \hat{w}_i \) and \( \hat{w}_j \) of different weights is zero.

**A strong near-optimality condition:** For any weight sequence, there exists no specialised scheme for sampling \( k \) items with unbiased weight estimators that gets smaller variance sum than priority sampling with \( (k + 1) \) items.

(Conjectured by DLT and settled by Szegedy (2006).)

**Performance in practice:** In the context of internet traffic analysis, outperforms all other sampling schemes by an order of magnitude.
Statistics and computer science share some common aspects.

- Motivated by down-to-earth practical considerations.
- Both subjects have a rich and deep theory.
- Major developments in both subjects have come in the twentieth century.
Concluding Remarks

- Statistics and computer science share some common aspects.
  - Motivated by down-to-earth practical considerations.
  - Both subjects have a rich and deep theory.
  - Major developments in both subjects have come in the twentieth century.
- Creative explorations of connections between the two subjects have been made in the past and continue to be made.
Thank you for your attention!