Advanced Mathematical Methods for Telemetry Analysis

Dr. Christopher Landauer
The Aerospace Corporation

Computer Systems Division
13 January 2015
2015 Spacecraft Flight Software Workshop
27 Oct 2015, JHU / APL

• Outline
  • Spacecraft Telemetry Issues
  • Software Caused Failures
  • Software Telemetry
  • Advanced Mathematical Methods
  • Some Example Applications
  • Some Practicalities
  • Conclusions and Prospects
  • Appendix: Definitions and Examples of Selected Methods

• Does not contain US Export Controlled information
Summary

• Spacecraft telemetry delivers an enormous amount of data.
  – Usually and mostly about the hardware.
• Advanced mathematical methods have the potential
  – To detect weak early indications or precursors of anomalies,
  – To identify unexpected correlations in system measurements,
  – To identify major hardware degradation in advance.
• These capabilities may lead to earlier and better behavior predictions
  – For ground-based analysis of spacecraft operation.
  – Eventually, maybe better on-board anomaly prediction / detection,
    • Even correction of some anomalies
• Experimentation is needed to understand what is and is not detectable.
  – How far in advance anomalies can be detected.
  – Which advanced mathematical methods are more effective.
  – What other data would be useful.

The big problem is software.
Spacecraft Telemetry Issues

• Classic analyses
  – Acceptable intervals and short-term trends

• Rarely
  – Cross-mission norms and anomalies

• Predictive models based on planned mission profile, and history of similar programs, satellites, or components

• In earlier papers,
  – We advocated extensive software telemetry.
  – We also advocate more telemetry for detailed instrumentation.

    • MUCH more!
    • We explain later how this choice need not be the bandwidth problem it might seem to be

Extend telemetry analyses to more software, cross-missions and longer times
Software Caused Failures

- NASA's Mars Climate Orbiter (23 September 1999):
  - Different developers used different physical units in thruster commands.
- NASA’s Mars Polar Lander (03 December 1999):
  - Software did not distinguish leg deployment from landing shocks.
- ESA’s Huygens Probe (14 January 2005):
  - One communication channel (of two) was never commanded to turn on.
- All three have something in common:
  - They are attributable to insufficiently careful software or software design.
    - In the ESA case, it was a CONOPS problem that more carefully designed software could have detected.
    - Something quite simple went wrong quite disastrously.
- The problem is that there are many millions of simple things that can go wrong.
  - Similar issues occur in some military satellite failures.
  - Descriptions of these ones are more widely accessible.

We have no development methods capable of managing this complexity.
Software Telemetry

• Each of these issues could have been found with more extensive software telemetry.

• Different context of reuse should have led to scrutiny of value ranges.
  – “It has flown successfully before” is a wholly inadequate justification.
  – Values can be reported and compared to expected value ranges.

• Simple on-board sanity checks should occur for all critical parameters.
  – This is where the numbers hit the processors.
  – This is where they should be checked.

• Ditto
  – Every value put into a database should be checked.
  – Telemetry should announce all values received prior to execution.

• Auditable chain of custody for every numerical parameter.
  – Not all problems are software.

These steps will help; they are far too seldom implemented, and by themselves, are not enough.
Advanced Mathematical Methods

• To provide perspective across:
  – Long time periods
  – Measurements
  – Missions

• Some example kinds of complex detailed analyses
  – Robust Statistics
  – Grammatical Inference
  – Event Pattern Correlation
  – Time Series Analysis
    • Hidden Markov Models
    • Dynamic Time Warping
  – Fractal Dimension
  – Dimension Reduction
  – Manifold Discovery
  – Topological Data Analysis

Many methods exist; we chose a few of them.
Some Example Applications

• Temporal characteristics detection
  – "Regime change"
    • Mission phases are planned well in advance
    • Expectable behavior can be predicted
      – Statistics can be predicted,
      – Different for each phase
  – Signal texture definition and identification
    • Expected noise level and pattern, general trend
    • Fractal dimension for change detection
  – Each hardware measurement has relatively well-known known degradation and failure characteristics.
    • Software is more difficult, because we know much less about degradation and failure of software components.
Some Example Applications (cont.)

- Correlations, expected and otherwise
  - Time delayed correlations can detect potential causalities.
  - Distinguishing causality from correlation depends on deviation correlations.

- Behavior in abstract trajectory spaces
  - Long term behavior of measurements over time and correlations thereof
  - Dimension reduction to identify nominal behavior and first excursions.
    - Computing expectable constraints
Some Practicalities

• Is this too much telemetry?
  – *We think the increased information outweighs the processing burden and decreases risk.*
  – *The increased information does not necessarily increase required bandwidth.*

• Compressed sensing for telemetry suites
  – *More telemetry, less bandwidth*
  – *Need to trust the compression process*
    • Process to detect measurement sequences outside the expected space
    • Also an occasional non-compressed item,
      – *used like tracers to verify compression process*
Some Practicalities (cont.)

- This is an example of dimension reduction
  - Define the expected space of measurement sequences
    - For each measurand separately (and some together)
  - Use manifold discovery to determine what part of the space actually gets used
    - This is also a prediction derived from nominal behavior
      - And expectable anomalies
    - Construct a low-dimensional manifold in the measurement sequence space
      - This is a common model with fewer parameters.
  - Map from actual measurement sequences to the smaller manifold
    - Fewer dimensions means fewer bits to send
      - Original measurement reconstructed from the values received
        - and the known common model
Conclusions and Prospects

• Exciting prospects for extracting more comprehensive information from telemetry
  – *Using a host of advanced mathematical methods (see the Appendix)*

• These computations need some serious experimentation for feasibility and location
  – *How much can be done on-board? on-line?*
  – *How much can only be done off-line or retrospectively?*
  – *How much isn't particularly helpful?*
  – *How much might be helpful with further development or research?*

• We need to collect much more extensive software telemetry for tracking the internal progress of computations.
  – *Happily, we can do that on the ground.*
Conclusions and Prospects (cont.)

• We need to experiment with software telemetry also
  – *What parts of the code are the most important to measure?*
    • decision points and other branches
    • function calls and parameters
    • interrupts and pre-emption
    • fixed point arithmetic computations
    • ...what else?

• We need new and much more extensive stress testing of units and components (to limit damage propagation or escalation).
  – *Can we prevent small errors from leading to catastrophic failure?*
  – *Can we prevent most errors from leading to catastrophic failure?*
  – *Study of error propagation through complex numerical calculations*
  – *Study of error propagation through non-numerical calculations*
Questions?
Appendix: Definitions and Examples of Selected Methods

• List of Selected Methods
  – Robust Statistics
  – Grammatical Inference, Event Pattern Correlation
  – Time Series Analysis, Hidden Markov Models, Dynamic Time Warping
  – Fractal Dimension, Large Dimensions
  – Dimension Reduction, Random Projections
  – Manifold Discovery, Local Linear Embedding, ISOMAP
  – Topological Data Analysis

• Define for each area
  – What it is about
    • Not formal definitions; those are available
  – What it can do
    • What it has already been used for, at least in a lab
  – What it might be able to do
    • With some extension and further development

This is a short informal introduction. Details can be found elsewhere.
Robust Statistics

• What it is
  – *Statistical computations that are not dependent on distribution assumptions*
  – *(every empirical distribution is usually assumed to be Gaussian; many are)*

• What it can do
  – *Identify outliers, identify vulnerabilities of any statistical summary*
  – *(what fraction of bad data can cause bad results)*

• What it might be able to do
  – *More reliable identification of noise (structured and not)*
  – *Better (less noisy) summaries of data*
Robust Statistics Example

- The example is about a sample set (where we consider different values for $x$)
  \[ \{0, 3, 7, 8, x\}. \]
- We can compute that the mean of this set is
  \[ m = 3.6 + 0.2 \times x, \]
- and that the variance is
  \[ v = 11.44 - 1.44 \times x + 0.16 \times x^2. \]
- As $x$ gets large, both of these increase without bound.

- The **breakdown point** of an estimator is the proportion (or number) of incorrect observations an estimator can handle before giving an (arbitrarily) incorrect result.
Robust Statistics Example (cont.)

• The robust "figure skating" rule is to omit the largest and smallest values.

• Then whenever x is larger than 8, the mean is 6 and the variance about 4.667
  – The breakdown point of this method is 1 if the statistic is one-sided
  – (a second very large value will be retained),
  – and the method can sometimes works with 2 outliers
  – (when one is excessively large and one excessively small).

• The influence functions for m and v are the algebraic expressions above, showing how m and v depend on x.
Grammatical Inference

• What it is
  – *Infer structure rules (syntax) from item sequences*
  – *Rules have the form LHS ==> RHS*
    • RHS is a sequence of terminal and non-terminal symbols
      – *terminal symbols correspond to items, so they do not change*
      – *non-terminal symbols correspond to classes of structured items*
    • LHS is also, with various restrictions that determine a classification
      – *PSG = Phrase Structure Grammar (at least one non-terminal)*
      – *CSG = Context Sensitive Grammar (one non-terminal in LHS changes to a sequence in RHS)*
      – *CFG = Context-Free Grammar (only one non-terminal)*
      – *RG = Regular Grammar (further restrictions on grammar)*
Grammatical Inference (cont.)

• What it can do
  – *Find compact rules for complex hierarchical behavior structures*
    • Structure recognition (parsing)
      – *Make sure the language contains enough correct sets*
    • Generation of random sets
      – *Make sure the language contains few incorrect sets*

• What it might be able to do
  – *New mathematics to define partially ordered set grammars*
  – *(see Event Patterns below)*
    • Inference of pattern grammars from complex data
Grammatical Inference Example

• Start with a large finite set of strings (sequences of characters)
• Problem is inherently ambiguous
  – *Many grammars generate languages that contain exactly the same strings*
  – *Usually want the smallest one*
• Sample inference rules (each makes a new non-terminal symbol and substitutes it into the strings)
  – *If A . B occurs a lot, infer new rule Z ==> A . B*
    • (and replace A . B with Z wherever it occurs)
    • (and replace A and B with Z wherever they occur)
    • (we can retain the context X _ Y in the rule if it is a CSG)
  – *If sequences of A's occur a lot, infer new rules Z ==> A and Z ==> Z . A*
    • (and replace sequences of A's with Z wherever they occur)
    • (or just Z ==> A+ if RG)
• Problem: cannot infer CFG with only positive examples
• Inferring recursion is difficult, but useful (it distinguishes CFG from RG)
Event Pattern Correlation

• What it is
  – Matching complex occurrence patterns separated in space and time
  – General partially ordered sets, instead of merely sequences

• What it can do
  – Make connections from similar structure

• What it might be able to do
  – Highly scalable algorithms for correlations among many sequences
    • Want something with complexity less than quadratic if possible
  – New mathematics of non-sequence correlation
    • Extension of previous algorithms for grammatical inference and parsing
  – Much more flexible event pattern detection and correlation
    • Then add a temporal component
Event Pattern Correlation Example

• Identifying TCP connection attempts
  – *Client side: Start in Initial state*
    • Send SYN
      – *SYN_SENT: Wait for SYN and ACK (or close from client)*
    • Receive SYN and ACK
    • Send ACK
      • Connection Established
  – *Server side: Go to LISTEN mode (Wait for SYN)*
    • Receive SYN
    • Send SYN and ACK
      – *SYN_RCVD: Wait for ACK*
    • Receive ACK
      • Connection Established
    • Receive RST, go to Initial

• These message sends can be detected
• Many coordination protocols can be identified this same way
Time Series Analysis

• This topic is too big, consider Hidden Markov Models (HMM), Dynamic Time Warping (DTW)

• What HMM is
  – Data implies maximum likelihood model parameters
    • Model implies sample data
  – EM algorithm is the iteration of this process

• What it can do
  – Determine certain kinds of hidden structure (HMMs are an example)
  – Impute missing data
  – Compute sensitivity of models to imputed data

• What it might be able to do
  – Algorithms to speed up EM iteration
  – Other hidden structures besides periodicity (via Fourier analysis)
    • Cascaded structures (e.g., periodicity depends on mode, mode changes according to Finite State Machine or Markov Model
Hidden Markov Model Example

• Long sequence of English text (words only)
• Assume two-state Markov chain and size 27 (letters + space) output alphabet
• Compute Maximum Likelihood estimate for transition probabilities and output maps
• Apply forward-backward algorithm (linear in the total length of the string)
  – Compute transition matrix and output mapping that maximize the likelihood of the observed string
  – Then the states clearly correspond to vowels and consonants
    • Letters y, w, and r are slightly more ambiguous than others
• Similar results hold for up to 12 states
  – Phonetically meaningful categories derived from ordinary English spelling
Dynamic Time Warping

• What DTW is
  – *Matching one time series to another, with different and possibly variable time progressions*

• What it can do
  – *Match speech patterns to templates*
  – *Match business rhythms of different duration*

• What it might be able to do
  – *Extend to a plane to match images in a new registration style*
    • Local distortion
    – *Match computers running the same code by output signals or by heat generation / power use*
Fractal Dimension

• What it is
  – Determine fractal dimension of a set, which determines minimum guaranteed embedding dimension (using tuple map)

• What it can do
  – Bound the computation of complex dynamics
    • If d = fractal dimension,
    • then any m > 2 * d + 1 is sufficient for embedding the dynamics
    • (and smaller dimensions can work)

• What it might be able to do
  – More reliable computation of fractal dimension
  – Much faster computation of a close upper bound
  – Hard to spoof
Fractal Dimension Example

- Koch curve, dimension $\log 4 / \log 3 \sim 1.26186$
- The dimension of any *self-similar* figure can be computed fairly readily.

![Koch curve diagram]

level 0  level 1  level 2  level 3
Things are Tricky in Many Dimensions

• The n-volume $V_n$ of the unit n-sphere decreases rapidly to zero as n increases
  • $V_0 = 1$, $V_1 = 2$, $V_2 = \pi$, $V_3 = (4/3) \times \pi$ (initial values, known values)
  • $V_n / V_{n-2} = 2\pi / n$ for $n \geq 2$,
    – which ratio monotonically decreases for all $n \geq 1$,
    – and the n-volume $V_n$ decreases for $n \geq 7$ (after increasing for $n < 7$)

• If we inscribe an n-sphere in an n-cube, the n-volume ratio goes rapidly to zero
  – Eventually, other unit n-spheres can be packed into the corners disjointly
  – (Martin Gardner devoted an entire column to these issues)

• The volume of an n-annulus (n-sphere shell), with inner radius = 90% outer radius is almost the same as the volume of the n-sphere, and the ratio goes to 1 as n increases
  • ratio = 1 - (0.9) $^n$

Almost everything happens at the boundaries or in the corners.
Dimension Reduction

• What it is
  – *Structure preserving map from high-dimensional space into low-dimensions*
  • Different structures need different algorithms to compute different maps
    – *Nearby pair distances*
    – *Local linear relations*
    – *Geodetic distances*
    – *Tangent space alignments*
  • Generally choose how many dimensions in advance
    – *and see later how well it worked*
  • Could conceivably use an intrinsic dimensionality like Fractal Dimension
Dimension Reduction (cont.)

• What it can do
  – *Reduces computational effort, sometimes a lot (millions or more dimensions to hundreds or even just dozens)*
  – *Images, video and other signals, abstract trajectories and other event patterns are all high-dimensional data*
  – *Relationship and other connection patterns are high-dimensional data*
  – *Trajectories of other values can be considered as elements in a high-dimensional abstract trajectory space*

• What it might be able to do
  – *All of these can be simplified with some structure preserved*
    • Usefully? We don't know yet
  – *More different kinds of preserved structures*
  – *Data-dependent algorithm choice*
  – *Faster and more robust algorithms (better source data uncertainty management)*
Dimension Reduction Example

• Non-linear map (J. Sammon)
• Given a point set \( X \) in \( \text{Real}^D \) for large \( D \), we want a mapping
  • \( f : X \rightarrow \text{Real}^d \)
    – usually with \( d=2 \) or \( d=3 \) and minimum \textit{distortion}
• Compare \( d(f(x),f(y)) \) to \( d(x,y) \) over all pairs \( x, y \) in \( X \)
  – \textit{Distortion} is sum of squared distance differences
    • \( (d(f(x),f(y)) - d(x,y)) \sim 2 \)
    – Normalized by original distance \( d(x,y) \) to emphasize preserving small distances
    – Each point in \( X \) is mapped separately, so there are \( d \times |X| \) variables
• Simple (if time-consuming) optimization problem
  – Many global minima (every permutation of \( X \), every isometry of \( \text{Real}^d \)), so we need some criterion to limit translations
    • Maybe one point is origin, always mapped to 0
    • Maybe sum of images is 0
    • Maybe choose \( d+1 \) points to anchor to a particular isometry in \( \text{Real}^d \)
Some Theorems

• Johnson-Lindenstrauss Theorem

  – *Any n point set in any Euclidean space can be embedded in k dimensions*
  
    • *k = O(log n / ε^2)*
    
  – *without distorting any pair distance by more than a factor of*
    
    • *1 ± ε*
    
  – *for any ε with 0 < ε < 1.*

  – *The issue is to find the smallest useful value of the constant for k.*

  – *The weird part is that Random Linear Projections almost always work.*
    
    • There is lots of research on how well restricted classes of projections work.
Some Theorems (cont.)

- Concentration of Measure Theorem

  - Write $S^{n-1}$ for the (n-1)-sphere in n-space, $\{ x \in \text{Real}^n \mid |x| = 1 \}$.

  - If $f : S^{n-1} \rightarrow \text{Real}$ is 1-Lipschitz
    - (i.e., $|f(x) - f(y)| \leq |x - y|$ for all $x, y$),
    - and if $s()$ is a normalized measure on the sphere
      - (i.e., $s(A)$ is the ratio of the area of $A$ to the area of the sphere),
    - Then
      1. There is at least one real number $M$ (called a median for $f$) for which
        - $s(\{ x \mid f(x) \leq M \}) = 1/2 = s(\{ x \mid f(x) \geq M \})$,
      2. $s(\{ x \mid |f(x) - M| > \varepsilon \}) \leq 2 \times \exp(-n \times \varepsilon^2 / 2)$
        - (i.e., $f$ is very seldom far from $M$, so it is almost constant)
Manifold Discovery

• What it is
  – *Identify low-dimensional manifold in high-dimensional data (example of dimension reduction)*

• What it can do
  – *Discover unexpected and previously unknown constraints*
    • Local linear structure
    • Other neighborhood properties
    • Connected distance properties

• What it might be able to do
  – *Better management of data content and noise reduction*
    • Compressed sensing
  – *Dynamic characteristics of high-dimensional images and signals*
    • Track changes, notice trends, identify fluctuations
  – *Allows interpolation for new data points*
  – *Hard to spoof*
LLE, ISOMAP

• Two methods for Dimension Reduction
• LLE = Local Linear Embedding
  – Local structure is defined by the best linear approximation of a point by its near neighbors, recorded in a coefficient matrix
  – Then the projection is the set of points that best preserves the coefficient matrix
• ISOMAP
  – Local structure is defined as the graph of its near neighbors;
    • Distance is measured in edge lengths
• There are many others; all depend on a choice of neighborhood
  – Distance or count threshold for size of neighborhood
  – Few or no studies of the effects of such choices
• Concentration of measure theorem makes all distance-based methods suspect
  – Examples can be found in the literature
Topological Data Analysis

• What it is
  – Fast homology computations for complex high-dimensional structures
  – (mapping complex geometric and topological structures to computable algebraic structures)

• What it can do
  – Summarize complex connectivity structures at all scales, robust to noisy data

• What it might be able to do
  – Compute sensitivity of structure to noise level
  – More effective image pyramid (less compression in slowly varying parts of the image)
  – Detect fundamental change in organizational relationships and other structures
  – Detect subtle differences among correct, erroneous, and fake data
  – The "tidy set" is a significant computational reduction (size and therefore time).
Second List

• The lead candidates for a next round of study are:
  – Uncertainty propagation
  – Identification of missing data and its potential utility to guide collection requests
  – Object extraction from images and signals
  – Flow mapping and non-explicit differential equations
  – Natural Language Processing (in multiple languages)
  – Graph theoretic analyses
  – Spatial-Temporal Reasoning
  – Advanced steganography

There are, of course, many potentially useful methods we have not listed here.
Further Information

• For questions and / or comments, please contact the author

chris.landauer@aero.org