DECISION MAKING IN SCIENTIFIC MACHINE LEARNING

Mark Temple-Raston

Decision Machine, New York, USA

ABSTRACT

Scientific Machine Learning is built on the science-of-counting, is deductively solvable, and well-suited to business and human applications that naturally count. From the Gibbs formalism, Scientific Machine Learning produces unique and exact scientific measurements that define the state of the time-series. Time-series itself defines a geometric structure tailor made for prediction, optimization and decision making. Inventory management decisions will demonstrate Scientific Machine Learning without introducing models or model bias.

KEYWORDS

Machine Learning, Supply Chain Management, Prediction, Optimization, Interaction, Information Theory, Decision Theory

1. INTRODUCTION

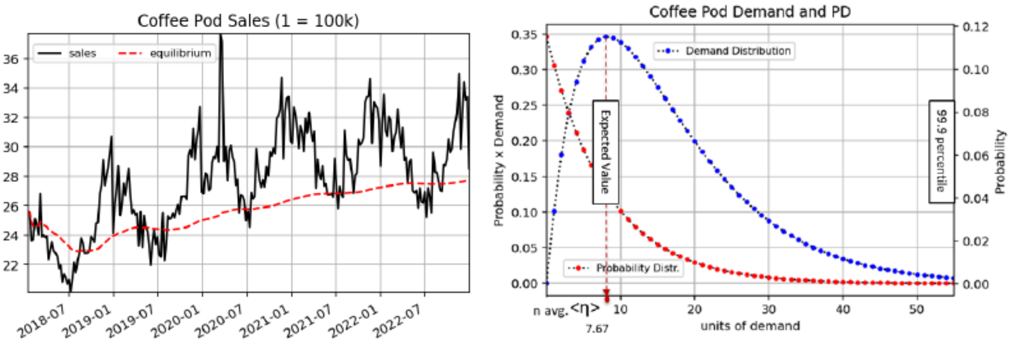
The science-of-counting (SoC) is founded on the principle of maximum information entropy, a method of logical inference used to define both science and machine learning through enforced constraints on information [1]. A person's confidence or certainty in counted data can be expressed mathematically, where the counted data are the enforced constraints on maximum entropy. So that, 23 refrigerators are exactly 23 refrigerators; no plus-or-minus allowed. The simplest, unsupervised, non-trivial and solvable machine learning is built on the science-of-counting, appropriately called Scientific Machine Learning. Scientific Machine Learning (SML) ingests time-series and returns scientific measurement. Deductive decisions are layered on top of SML, for example, dynamic generalizations of N-Armed Bandits [3]; exact and unbiased inventory levels and resource allocations; and, where these same decisions can be both complex and competitive.

Emerging in the first half of the 19th century, the science-of-counting generated new and fruitful scientific disciplines (thermodynamics, the theory of electricity, physical chemistry) to generate many successful practical applications for the industrial revolution. Unlike mechanics (constant total energy and no external interactions), the science-of-counting applies to all time-series, mechanical or otherwise, that count states, events, or a unit of measure, as natural number multiples (1,2,3,...). SML integrates time-series to provide the scientific justification and geometric structure for a fully deductive and dynamic Machine Learning!

Scientific Machine Learning is illustrated in the Introduction with six years of weekly recorded sales time-series for a Consumer Product Good (CPG)---individual-serving coffee pods. The sales data is plotted in Figure 1a (black curve). The arithmetic mean (average) of the sales data is superimposed on Figure 1a (dashed red curve). When the sales data and arithmetic mean overlap, the sales data behaves statistically, that is, close to equilibrium (sales are just as likely to go up, as

DOI:10.5121/mlaij.2026.13101

to go down). Sales data above the average and below the average behave non-linearly with additional forces and heat dissipation that affect market behaviour.



(a) Coffee Pod Sales

(b) PD and Demand for Coffee Pods

Figure 1: Probability and Demand Distributions for Sales

Scientific Machine Learning ingests the sales data and returns scientific measurements, including the exact probability distribution (PD, red dots) in Figure 1b. Because counting is exact, the SML probability distribution consists of discrete points, corresponding to each natural number multiple of the unit one for demand. Note that the countable, infinite sum of points in the probability distribution add to one---not the area under the curve. Exact probabilities determine precisely the boundary for the 99.9 percentile, which tends to be much closer to equilibrium than statistics would imply, and gives means to thicken tails. The demand distribution (blue) is the direct multiplication of the probability with the unit multiple and therefore is also discrete. The countable, infinite sum of the points in the demand distribution add up to the expected demand.

We refer often to the exact probability distribution (PD) in Figure 1b, because of its utility, and return to the Coffee Pod data for illustration. In the next section, we develop Decision Machine on top of the Scientific Machine Learning engine and explore essential capabilities (each subsection).

2. Decision Machine

2.1. Inventory Management

The expected total inventory cost starting with an initial supply of n ϵ **N** items is denoted by f(n). By enumerating the various cases that correspond to excess demand over supply, and to be able to fulfil the demand, the minimum expected total cost, f(n), is given by [2]:

$$f(n) = \min_{y\geq n}\left[k(y-n) + a\sum_{y=N}^{\infty} p(n-y)\varphi(n) + a\sum_{y=0}^{N-1} f(y-n)\varphi(n) + af(0)\sum_{y=N}^{\infty}\varphi(n)\right], \quad (1)$$

Where y is the inventory level, and y-n is the difference between the stock level and what is currently in inventory, n. The cost of ordering z items to increase the stock is given by k(z). The cost of ordering z items to meet an excess of demand over supply is determined by the penalty cost, p(z). The discount ratio, a, is the time value of the product. The probability that the demand will be n is denoted by φ(n). Our objective is to calculate N in eq. (1) to minimize inventory cost. Bellman's functional equation (1) is solved exactly with SML in the special case that conforms with business practices, where the cost of ordering is directly proportional to the amount ordered, so that k(z)=kz and p(z)=pz. Assume that a, k and p are fixed, and that there is no shipping delay, d=0 (instantaneous fulfilment). The minimization of eq. (1) gives

$$\varphi_{d=0}^{N+1} = \frac{k(1+a)}{a(p-k)}. \tag{2}$$

Recall that SML processed the coffee pod sales time-series in the Introduction to give an exact discrete probability distribution (Figure 1b). Therefore, $\varphi_{d=0}(n)=p(\lambda,E)(n)$ for the demand, which can then be inserted into eq. (2). Solving for N:

$$\lfloor N \rfloor = \frac{\ln(k(1+a)) - \ln(a(p-k))}{\ln p(\lambda, E)} - 1. \tag{3}$$

The floor function truncates N to a natural number. For the penalty ratio p/k = 5 and a=0.99, the optimal inventory level is calculated from the historical time-series and plotted (blue curve) in Figure 2.

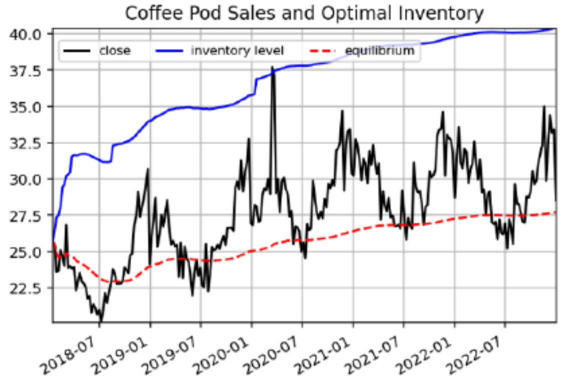


Figure 2: Optimal Inventory Level for $p/k = 5$

The same calculation applies to the time-series for a shelf of product. Seasonality in inventory levels can be implemented; in the next section, seasonality in sales data is examined. Finally, from the prediction eq. (8) in the next section, the future probability distribution $\varphi_d(n) \equiv p_d(n)$ is computed, and the inventory level eq. (2) is seen to hold for replenishment times, d > 0.

2.2. Prediction and Optimization

The previous subsection solved an inventory problem using the exact probability distribution defined by a time-series. However, there is a deeper reason for our ease: the science-of-counting defines a discrete geometry that tells us how energy, mass, momentum, and so on, must fit together. Time-series in SML defines a discrete geometric structure, the connection one-form, A. When we add a unit decision direction, X, then the following geometric relation holds:

$$p_{i+1} = p_i \exp(A(X_i)). \tag{4}$$

Equation (4) defines the geometric dynamics for the probability distribution in SML for solutions to prediction and optimization.

Two equally important scientific quantities emerge when time-series data is processed by SML (via Gibbs): the expected value <n>, and the expected strain, <ϵ>. However, when energy enters or exits a time-series (E ≠ 0), the natural coordinates are instead the expected demand, <η>, and the expected supply, <ξ>, connected by the coordinate chart, T, between the two coordinate systems:

$$\begin{bmatrix} <\eta> \\ <\xi> \end{bmatrix} = \begin{bmatrix} 1 & E \\ E/p^2 & 1 \end{bmatrix} \begin{bmatrix} <n> \\ -\lambda <\epsilon> \end{bmatrix} \equiv T\mathbf{x}. \tag{5}$$

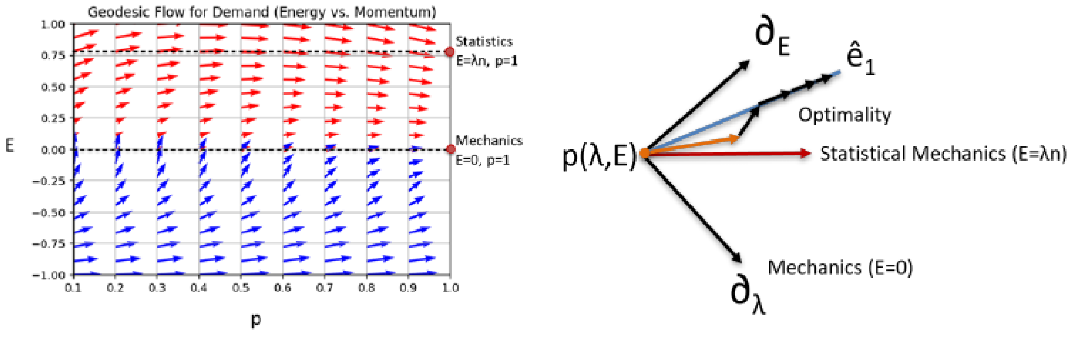
The self-coupling, λ, enforces the SML counting constraints. When the displacement energy vanishes, E=0, the expected supply and demand reduce to the Gibbs coordinates and statistical mechanics. The eigenvalues (λ+, λ-), and eigenvectors (ê1, ê2), for the chart E are easy to calculate. In the eigenbasis for the matrix, T is diagonalized and the matrix equation decoupled, so that future demand and supply are given by a sequence of shared energy-momentum ratios, (E_0/p_0, E_1/p_1,...,E_d/p_d), equivalently, the percentage increase or decrease, r = E/p, at each stage:

$$<\eta>_d = \prod_{k=0}^{d} \left(1 + r_k\right) <n>, \tag{6}$$

$$<\xi>_d = \prod_{k=0}^{d} \left(1 - r_k\right) (-\lambda <\epsilon>), \tag{7}$$

$$p_d = \prod_{k=0}^{d} \left(\frac{1 + r_k}{1 - r_k}\right) p. \tag{8}$$

For constant growth, r for all k, the trajectory for the expected demand is given by eq. (6) --- a straight line. See also Figure 3b. Any decision vector X can be written as a linear combination of eigenvectors, X = x ê1+y ê2, so that the connection can be evaluated on X, A(X) = x A(ê1) + y A(ê2). The time-series connection on the demand eigenvector is substitute into eq. (4) to calculate the probability-momentum and displacement energy for the next time-step, p_{i+1} and E_{i+1}, respectively. The system now evolves on its own, to give the geodesic flow in Figure 3a. Statistics assumes constant energy (Figure 3a), conventionally achieved by coupling the system to a constant temperature heat bath, that provides the needed energy. Fig. 3a tells us what happens without the heat bath.

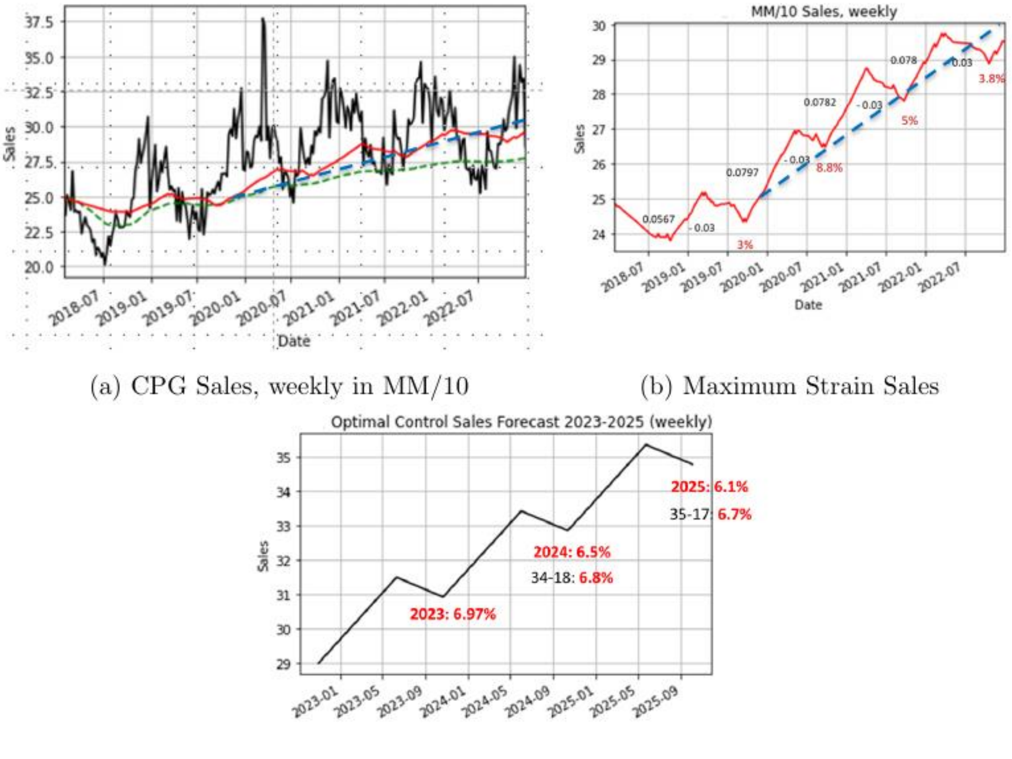


(a) Geodesic Flow for Expected Demand (b) Bellman's Principle of Optimality

Figure 3: Creating an Optimal Policy

The Expected Demand eigenvector, ê1, is the straightest, flat curvature direction defined by a constant growth rate, r; geometrically flat, but not a geodesic, because increasing energy must enter the system to achieve optimality. An optimal policy requires that whatever the initial state and initial decision direction, over which we may have no control, the remaining decisions must constitute an optimal policy. The exact increase in energy needed to follow the Expected Demand eigenvector can be calculated from Figure 3a and 3b to achieve the greatest increase in demand for the least energy in Figure 3b (optimality), and the dashed blue lines in Figure 4a and Figure 4b. Diminishing return for the expected demand is due to increasing strain due to the state equation.

Of course, there is no obligation to minimize energy expenditure as we did with the dashed blue line; the budget could be bigger. The red demand paths in Figure 4a&b are measurements from the sales data using SML. In plots of demand under strain, linear elements are observed, which maximize the expected demand and the expected supply, the maximum extension for the given energy. The linear behaviour observed experimentally is expected at second order (statistics with constant drift). The piece-wise linearity in the red path also reflects seasonality in sales: there is a high season and low season, with durations defined by sales operations in weeks that add up to 52. The slopes of the triangular demand curve are determined from our plots for the historical time-series (see Figure 4b). Using the slopes calculated for 2020 for future returns, impressive year-over-year percentage increases in sales for the red demand paths are calculated (see Figure 4c): 2023: 6.97%, 2024: 6.8%, 2025: 6.7%.



(a) CPG Sales, weekly in MM/10

(b) Maximum Strain Sales

(c) Maximum Strain Forecast

Figure 4: Max Improvement

2.3. Decision Optimization

The previous sections calculated the inventory level and control actions for an individual product given sales history time-series: the exact discrete probability distribution (Figure 1b) and the probability dynamics implied by eq. (4). In this section competition and constraints on decisions are developed. Consider two SKUs (stock keeping units), A and B, with probability distributions, p_A and p_B, demands, <η_A> and <η_B>, measured from time-series, and with resupply times, d_A and d_B, respectively. We allow direct costs, penalty costs and discount ratio (k, p, and a) to be different for each SKU. If, for technical reasons, only one of the two SKUs can be resupplied at a time, which should be resupplied first to minimize expected cost? The expected cost for each SKU, <C_A> and <C_B>, is given by:

$$< C_A > = p_A^{N_A} \quad < \eta_A >, \tag{9}$$

$$< C_B > = p_B^{N_B} \quad < \eta_B > . \tag{10}$$

Because in this case only one SKU can be resupplied in the current time-step, the optimum decision is the SKU with the least expected cost, either SKU A (eq. (9)) or SKU B (eq. (10)), the minimum of <C_A> and <C_B>.

Construct a Decision Matrix, D, that systematically evaluates each demand and all decisions simultaneously, and selects the best decision that minimizes expected cost. To illustrate with a third decision point for the current pair of SKUs, where the direct costs, k, to resupply both SKUs without penalties is less/smaller, due to shipment bundling SKUs together more efficiently, purely hypothetical. This decision point is added to produce the Decision Matrix below that connects measured expected demands (time-series), <η_A> and <η_B>, to decisions based on expected cost, <C_A>, <C_B>, <C_{AB}>. Again, identify the optimal decision with the smallest expected cost.

Decision Matrix

$$\text{Decisions} \quad \begin{bmatrix} < C_A > \\ < C_B > \\ < C_{AB} > \end{bmatrix} = \begin{bmatrix} p_A^{N_A} & 0 \\ 0 & p_B^{N_B} \\ p_A^{N_A} & p_B^{N_B} \end{bmatrix} \begin{bmatrix} < \eta_A > \\ < \eta_B > \end{bmatrix} \quad \text{Demands}$$

Each decision block is processed using time-series for the demands, and a configuration file that initiates the processing and defines the contractual terms and shipping times (kpad) and the Decision Matrix, to generate the lowest cost inventory levels subject to constraints. The Decision Matrix in Decision Machine can be generalized to any finite number of demands defined by time-series, and a finite number of decisions, without model, bias or parameter fitting.

2.4. Interactions

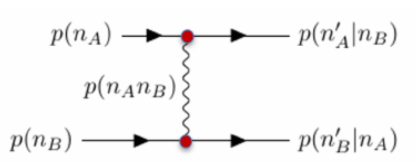


Figure 5: Interaction Diagram for SML

The information Lagrangian for the science-of-counting evidently generalizes to include two time-series with interaction:

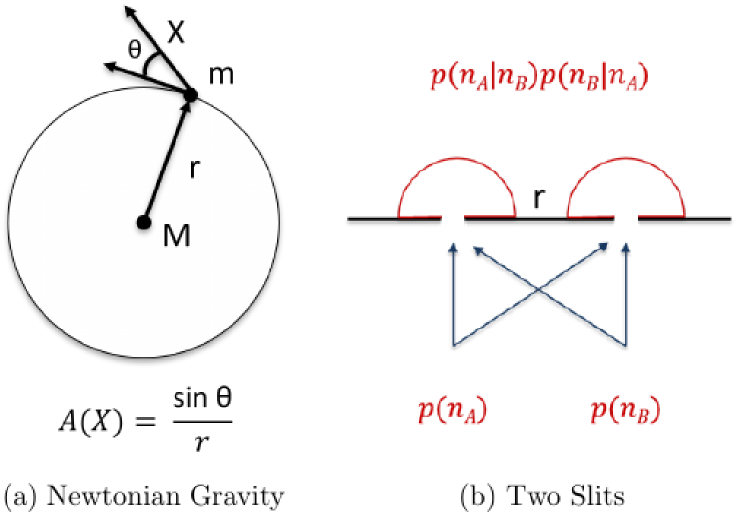
$$\mathcal{L} = \mathcal{L}_A + \mathcal{L}_B - \lambda_{AB} < n_A n_B > - \sigma_{AB} < v_A v_B > . \tag{11}$$

Energy is allowed and observed to enter and exit each time-series. When pairwise interactions are included (so that λ_{AB} ≠ 0), the science-of-counting determines an algebraic relationship between the pure, joint and conditional probabilities in Figure 5, given by:

$$p(n_A n_B)^2 = \left[1 - \frac{p(n_A)p(n_B)}{p(n_A n_B)}\right] p(n_A)p(n_B) \tag{12}$$

$$= \Big(p(n_A|n_B)p(n_B|n_A)\Big)\ p(n_A)p(n_B). \tag{13}$$

A special case of the interaction diagram in Figure 5 occurs when all masses in the momenta are taken to be constant, as it is in Newtonian Gravity. When masses and the distance between the masses are counted and constant, the necessarily circular geometry of Newtonian counting (see Figure 6a) is derived from the geometric equation of motion for momentum. The remaining Newtonian conics can be similarly formulated. The geometry of the connection determines the curvature and the forces (resistance and stress) that drives behaviour in Newtonian Gravity. Returning to entanglement, time-series measurements give an exact value for the entangled state p(n_A|n_B)p(n_B|n_A) post-interaction, so that if we knew one leg of the output perfectly, then we would know the other (classical EPR). In practice, however, we don't know either output leg exactly, so that classical entanglement remains. The two-slit experiment (see Figure 5b) is another example.



(a) Newtonian Gravity

(b) Two Slits

Figure 6: Two Examples in Classical Dynamics

3. Conclusions

We take pleasure in how accessible calculations in Scientific Machine Learning are. With only an undergraduate's knowledge of eigenvalues and eigenvectors, the meat of the science-of-counting---the predictive dynamics---can be figured out for yourself. While non-trivial, the calculational busy work is modest because the matrix is only two-by-two!

It is too early to draw conclusions, but it seems sure that neglecting scientific measurements in decision making can only lead to a deteriorating operational environment. We are, in some cases, responsible both for the choppy seas and rocking-the-boat, that cause our ship to swamp and sink beneath the arithmetic mean. There is room for the restraint that improves performance.

REFERENCES

[1] Jaynes, E.T. (2003) Probability Theory: The Logic of Science, Cambridge University Press. (Chapter 11, especially)

[2] Bellman, Richard (1957) Dynamic Programming, Princeton University Press.

[3] Temple-Raston, M. (2025) Scientific Machine Learning, Computer Science and Information Technology (CS & IT), Vol. 15, No. 18. (https://ssrn.com/abstract=5591450)

AUTHOR

Mark Temple-Raston, PhD, is a seasoned innovator and executive with over two decades of global experience in information technology and financial services. He is co-founder of Decision Machine, a company that applies a scientific approach to machine learning to deliver unbiased, precise analyses for market science and decision-making.

Before launching Decision Machine in 2015, Mark spent 20 years on Wall Street, holding senior leadership roles at Citigroup, where he contributed to Global Functions, Enterprise Architecture Governance, and the Chief Data Office. His expertise spans enterprise architecture, data governance, and risk management across industries including healthcare, logistics, and aerospace.

Mark earned his doctorate in Applied Mathematics and Theoretical Physics from the University of Cambridge, UK. His work reflects a commitment to rigorous, transparent systems that blend mathematical precision with practical business and human insight.