Understanding KPI trade-offs - key challenges of modelling architectures and data acquisition
Gurtner, G. and Cook, A.J.

This is an electronic version of a paper presented at the Data Science in Aviation 2017 Workshop, Cologne, Germany, 29 September 2017.

Details of the workshop: http://datascience.aero/2017-workshop/

The WestminsterResearch online digital archive at the University of Westminster aims to make the research output of the University available to a wider audience. Copyright and Moral Rights remain with the authors and/or copyright owners.

Whilst further distribution of specific materials from within this archive is forbidden, you may freely distribute the URL of WestminsterResearch: (http://westminsterresearch.wmin.ac.uk/).

In case of abuse or copyright appearing without permission e-mail repository@westminster.ac.uk
Understanding KPI trade-offs

Key challenges of modelling architectures and data acquisition

Gérald Gurtner, Andrew Cook

29 Sept. 2017
EASA, Cologne
Vista - goals and objectives

Vista aims to study the main forces (‘factors’) that will shape the future of ATM in Europe at the 2035 and 2050 horizons

More specifically:

• trade-off between, and impacts of, primary regulatory and business (market) forces;
• trade-offs within any given period;
• trade-offs between periods;
• whether alignment may be expected to improve or deteriorate as we move closer to Flightpath 2050’s timeframe

Focus on five stakeholders: airlines, ANSPs, airports, passengers, and environment.
Vista - Project overview

Workflow:

• Build an extensive list of **business** and **regulatory** factors likely to impact the ATM system.

• Classify the factors: short-term/long-term, likelihood of occurrence, importance of their impact on the ATM system, etc.

• Build current and future scenarios.

• Building model requirements:
  • *consider as many (important) factors as possible in a flexible way*;
  • *produce level of detail required and achievable to capture relevant metrics*.

• Iterative model development in consultation with stakeholders.

• Trade-off analysis.
Vista – How to produce a trade-off analysis

Trade-off: inverse relationship between two indicators. When one improves, the other worsens.

Two types of trade-off:

- Correlation with time series:
  - Past time-series: usually not enough data for macro indicators
  - Future time-series: need a model
- Causal relationship: with a model.

What about a change in the system? How to compute the relationship between metrics in totally new environment?
How to produce quantitative knowledge for the future?

Current World → Microscopic Observation → Raw Data → Dimensionality Reduction → Data KPIs Current situation
How to produce quantitative knowledge for the future?
How to produce quantitative knowledge for the future?

Current World → Microscopic observation → Raw data → Dimensionality reduction → KPIs Current situation

Microscopic observation → Raw data

Correlation relationships → Extrapolation

Data KPIs Future situation

Extrapolation

Machine learning domain
How to produce quantitative knowledge for the future?

Current World

- Microscopic observation
- Phenomelagic observation

Raw Data

- (Causal) laws between variables
- Causal and Correlation relationships

Physical modelisation

- Raw synthetic data
- Dimensionality reduction

Physical modelisation domain

Data

KPIs

Hypothetic situation
Challenges in data acquisition in Vista

'Raw data':

• Format (and sometimes content!) not consistent over time and over bodies providing them
• Openness: rarely open, usually expensive, or simply hard to get with very convoluted rules.
• Quality: individual projects redoing over and over the same quality checks on the same datasets.
• As many procedure to acquire data as number of datasets (at least): financial for airports, financial for airlines, financial for ANSPs, schedules, flight plans, real trajectories, itineraries, fares, etc.

'Phenomenological laws':

• Coming directly from theory and or other machine learning studies.
• Assumptions sometimes not clear, validity subject to other checks on the system
• Can be completely wrong, whereas raw data can lie only where recorded incorrectly!
Scenario definition in Vista

Vista is a 'what-if' scenario analyser. Answers to:

- *What happens if I do this in the system?*

And not:

- *What will happen in 2035 or 2050?*

==> Scenario definition, where different external factors can influence the system. Aim is **not** to compute the likelihood of a given scenario.

Factors are subdivided into two main categories:

- **Business factors**: cost of commodities, services and technologies, volume of traffic, etc. => demand and supply
- **Regulatory factors**: from EC or other bodies, e.g. ICAO, => ‘rules of the game’

Use in particular the different targets and high level views of SESAR to have a idea of the possible values of the parameters.
Regulatory factors:

• Regulations affecting gate-to-gate phase
  • SESAR development and integration (RSI): e.g., SES, Common projects.
  • Performance based regulations (RPB): e.g., Performance review body
  • ANSP requirements (RAR): e.g., Common requirements

• Regulations affecting airports
  • Airport demand (RAD): e.g., slots
  • Airport processes (RAP): e.g., ground handling market
  • Airport access / egress (RAA): e.g., airport access policies

• Regulations affecting other areas
  • Other regulations (ROR): e.g., passenger provision schemes, emission schemes

22 factors in total

Some of the regulatory factors are enablers of business factors
Scenario definition in Vista

Business factors

• Factors affecting gate-to-gate phase
  • SESAR operational changes (BTS): e.g., Free-routing
  • Other operational and technical changes (BTO): e.g., Passenger reaccommodation tools

• Airport processes and accessibility
  • Airport access / egress (BAA): e.g., multimodality
  • Airport processes (BAP): e.g., self-processing

• Demand and other economic factors
  • Demand evolution (BED): e.g., economic development
  • Other economic factors (BEO): e.g., fuel price

37 factors in total
Scenario definition in Vista

- Regulatory foreground factors
- Business foreground factors
- Regulatory background factors
- Business background factors

Foreground factors

Background factors

Scenarios

Foreground

Background
Multi-layered architecture of Vista

- **Foreground Factors**
  - Economic Model
  - Schedule mapping
  - Initial Mobility State

- **Scenario**
  - Exogenous Variables
  - Pre-tactical Layer
    - Passenger assignment
    - Flight plan generation
  - Learning Loop
    - Comparison with targets
    - Setting new initial state
  - KPIs

- **Background Factors**
  - E.g. regulations, technologies, forecasts
  - E.g. (near-final) capacities and demand
  - E.g. uncertainty, cost of delay, reaccommodation rules

- **Tactical Layer**
  - Mercury (mobility model)
  - Final Mobility State
  - Impact Trade-offs

Data science in aviation 2017 workshop, 29SEP17, EASA HQ, Cologne
Multi-layered architecture of Vista

Three main layers correspond to:
• Strategic:
  • Producing main flows in Europe based on macro-economic variables
• Pre-tactical:
  • Producing flights plans (and disruptions).
• Tactical:
  • Simulating a real day of operation with microscopic pax tracking.

Transversal layers consist of stakeholders:
• Airlines: choose flights, react to delay, etc.
• Airports: deliver departure and arrival capacity, create congestion, etc.
• ANSPs: deliver ATC capacity, create regulations etc.
• Passengers: choose best itineraries based on fares and other parameters, make their trips with possibility of disruption, etc.
• Environment: is passively impacted by NOx and CO2
Multi-layered architecture of Vista

Factors:
1. Factor 1: Val1, Val2, Val3, Val4
2. Factor 2: Val1, Val2
3. Factor 3: Val1, Val2, Val3
4. Factor 4: Val1, Val2, Val3, Val4

Model:
- Strategic Layer
- Pre-tactical Layer
- Tactical Layer

Initial Mobility State
Adjusted Behaviour
Learning Loop
Comparison with targets
Setting of initial state
KPIs
Final Mobility State
Impact Trade-offs

Data science in aviation 2017 workshop, 29SEP17, EASA HQ, Cologne
Data management in a multi-layered architecture

Data need to be:

• Consistent among layers,
• Easily accessible (for computing power),
• Traceable between the different blocks

==> All data are based on a single database, accessible to all the blocks. This ensures consistency, traceability and reproducibility.

More challenges come with this data architecture:

• How to enforce consistency between input and output of two block?
• How to take into account the multiple runs of the stochastic layers?
• What is the right balance between flexibility (NoSQL) and consistency (SQL)?

==> Now use a MySQL database.
Calibrating the model

Calibration is done in several steps:

• **Direct calibration:**
  • Extract some values from historical data (including literature) and set them directly in the model:
    • E.g.: price elasticity for passengers
  • Put some phenomenological relationships obtained otherwise:
    • E.g.: cost of delay for airline as a function of delay.

• **Indirect calibration:**
  • Supervised learning: a parameter is swept (in a smart way) in order for another one to reach a value extracted from data.
    • E.g.: cost of capital for airlines is calibrated to have the historical flows of passengers between airports.
  • Reinforcement learning: for instance, agents in the model modify their behaviour in order to be self-consistent across layers.
    • E.g.: cost of delay used to compute main flows should be the same as the actual cost of delay during the tactical phase.
Studying the output: how to recognise a trade-off?

- Stochastic context, correlative trade-offs
Studying the output: how to recognise a trade-off?

- Stochastic context, trade-offs comparison
Studying the output: how to recognise a trade-off?

- Deterministic trade-off: dependence of distribution over deterministic parameter
Trade-off example: predictability vs punctuality

- One airport, unpredictability of departure delay is changed artificially

![Graph showing the relationship between unpredictability and average delay. The graph indicates a downward trend as unpredictability increases.]
Trade-off example: LLC vs trad

- Simplified setup: four airports, two airlines LLC/trad, capacity increase of airport 3
Trade-off example: LLC vs trad

- On average, everyone is better off after the capacity increase
Trade-off example: LLC vs trad

- But some agents are actually losing from the capacity increase!
Conclusions

• Vista aims at understanding the trade-offs (or synergies) between KPIs in (current and) the future (2035 & 2050) air transportation world.
• Requires forecasting the values of the KPIs, and also their relationships:
  • Either by pure machine learning.
  • Or by injecting other phenomenological laws into the model.
• Vista is based on a multi-layered architecture requiring very diverse types of data as input.
• Additionally, the different layers of the model need to communicate smoothly and reliably, thus requiring a central data repository.
• Calibration (or training) is a main issue in this type of model and requires several steps involving data reduction and internal optimisation.
• The trade-off analysis requires different techniques, including statistical regressions, and also careful data aggregation. Different tools can be used to help choose the best situation, including Pareto analysis etc.
• Trade-offs can appear between different types of stakeholders, among different actors of the same type, among periods, etc.