Rethinking Assurance

Definition

Assurance is a broad term covering a wide range of applications, including: financial audit, application development quality, insurance *(assurance is the compliment of insurance)*, and manufacturing production quality. *Best Practices* leverage a shared *body of knowledge.*

Since we focus on business transactions in this course, our definition can be refined as: the ability to explain the behavior of transaction environments *(i.e., "obtain sufficient understanding and evidence").* And since our primary tool for explaining behavior is modeling, we refine this definition to: a set of ***practices*** for ***modeling*** the ***transaction generating processes*** and ***outcomes*** of a business.

Of course, the more our models explain, the less substantive testing we have to do, and the more efficient the audit *(and the happier the client). S*o, we want to leverage best practices for building these models, including:

- Measurement of ***credibility*** in ***probabilistic terms.***
- Quantification of ***beliefs*** *(i.e., encapsulation of experience and knowledge).*
- Measurement of ***effects*** *(i.e., influence)* of across transaction entities and states***.***

All with a focus on model quality, relevance, efficiency and application scope.

Technology Trends and Assurance

Building models that explain transactions is both enabled and challenged by an intimidating array of emerging technologies that will impact Assurance. One way to group these is by services:

- **SaaS** *(software as a service)* describes the migration of applications to the "cloud" *(e.g., a company moves accounting from an "on-premise" instance to SAP's cloud - reducing operations and maintenance, but giving up functionality because the cloud versions are standardized)*. This is driving standardization of many processes, reducing complexity and increasing efficiency and reusability of assurance practices and models.

  But, this trend also pushes differentiating processes *(a driver of business margins)* to point solutions *(more functionally specific business applications)*, increasing diversity in application portfolios, creating complex integration layers, and migrating logic and controls down level.

- **PaaS** *(platform as a service)* describes the utilization of cloud platforms to manage transactions and data *(like we do in class when we use analytics platforms like Azure Machine Learning)*. This is providing new analysis and data platforms - expanding the power and scale of modeling, enabling more effective and extensible approaches to assurance modeling.

  But, PaaS also enables algorithmic transaction creation and routing, distributed ledger exchanges, real-time IOT connections, etc., expanding transaction management into high-volume environments with complexity beyond current assurance practices.

Adoption of the enabling technologies by assurance teams has been predictable:

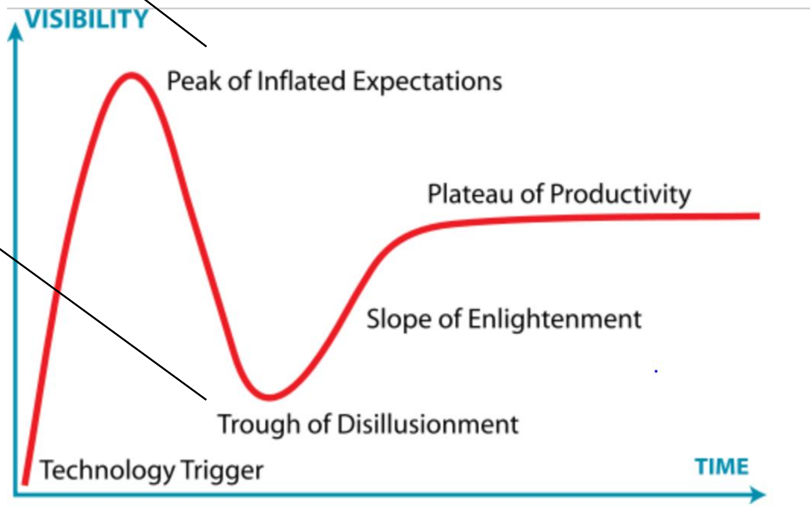
We now approach in the ***Peak Inflated Expectations*** where we see suboptimal application *(e.g., Audit "dashboards" that are low-dimension, non-parametric visualizations connected to larger warehouses – basically, repurposed 90's BI).* And "big data" and blockchain expectations are almost chimerical.

As we progress, the ***Trough of Disillusionment,*** will yield realizations that:

- Low-dimension visualizations are rarely adequate for inference
- *"Big Data"* platforms will not be able to consolidate diverse transactions without *"Big Transformation"* effort
- Distributed transaction ledgers are limited in processing throughput performance and logic implementation to see extensive deployment.

And of course, the technology will be blamed.

Eventually, we will move on to the ***Slope of Enlightenment*** *(the rest of the course anticipates this path)*



Gartner Research

BAUER COLLEGE OF BUSINESS
UNIVERSITY of HOUSTON

Levelset - Current Practices

Reviewing key Audit Practices:

- ***Sampling.*** The PCAOB allows both Statistical and Non-Statistical sampling in Tests of Controls and Substantive Testing *(AS 2315). Auditors are required to either use statistical sampling of transactions, or to apply statistical methods to determine sample size*[1]. However the sample size is determined, the application of *haphazard* or *judgmental* sampling methods results in a *non-statistical* analysis.

- ***Professional Judgment.*** The application of "professional judgment" in sampling is addressed in AS2315 §02: "The auditor often is aware of account balances and transactions that may be more likely to contain misstatements. He considers this knowledge in planning his procedures including audit sampling". It also states that "the auditor will have no special knowledge about other account balances and transactions that, in his judgement, will need to be tested to fulfill his audit objectives. Audit sampling is especially useful in these cases". Judgment is more often substituted for sampling in Control Testing *(evaluations of ERP security and validation can often provide confidence in transactions)* [1].

- ***Evaluation and Inference.*** The PCAOB states in AS2315 §26: "the auditor should ***project*** the misstatement results of the sample to the items from which the sample is selected." *(we will call this **modeling and simulation** too)*. The most common methods of ***projection*** are ratio and difference, but application is inconsistent, and engagement teams have difficulty understanding how to treat deviations and misstatements, possibly failing to recognize that a projected error coupled with sampling risk might result in a material misstatement [1].

1. "Current issues in Auditing" AAA Vol 9, Issue 2

Current Practice Weaknesses and Improvement Opportunities

- **Sampling.** Current practices assume that transaction populations fit *normal distributions*, predetermining sample strategy and population parameters $N(\mu,\sigma)$. That's not always a good assumption. Transaction distributions are often skewed, mixed models, etc. - so beyond sample size determination, representative sampling and projection require a different set of parameters *(e.g., λ, δ, α for skew normal)*. And transaction ledgers are almost always multilevel *(e.g., account/org/facility/project…)* with fixed, random and pooled effects that determine parameters. Transaction models should consider multiple distributions, with effects at multiple levels.

- **Professional Judgment.** Current practice relies heavily on professional judgment *(beliefs)* for determining procedures, but these beliefs are usually not quantified. Beliefs can be quantified and validated within statistical models. For example, ones' belief *(often based on prior evidence) in a parameter,* α *(e.g., correct classification %)*, *and confidence in that belief (variance σ^2 of α) can be integrated with prior and current data to project the probabilities of parameter values (classification %), and statistically valid* confidence levels can be determined.*

- **Evaluation and Inference.** Bayesian modeling can integrate beliefs and data to produce interpretable parameters which can be used to generate projections. Moreover, these models can explain a larger proportion of transactions than traditional models, reducing the need for substantive testing.

**Bayesian statistics is based on formal statistical methods, with an established base of critical application (e.g., FDA clinical trials)*

Future Challenges *(issues touched on technology trends section – worth extending)*

Blockchain

There are 2 big challenges:

- **Functionality.** Smart contracts have major limitations with encapsulating business logic (*e.g., pricing and order acceptance).* Additionally, most of this logic is already implemented in existing systems, so the benefit of migration is not compelling.
- **Scale.** Ethereum has reached ~ 3k transactions / sec and blockchain is significantly slower Most corporate performance benchmarks are ~ 5k/sec. These platforms are limited by their architecture and performance improvement is not likely.

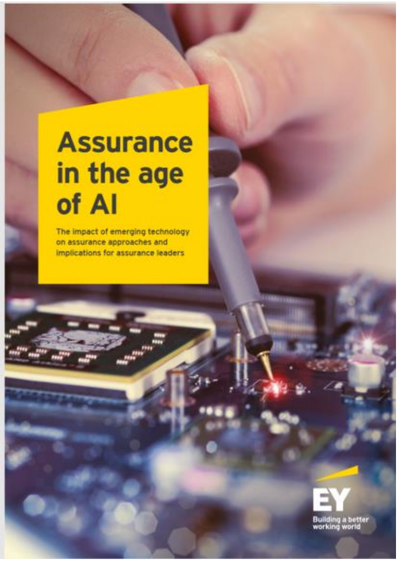
Application will most likely be limited to new partnerships involving low volume, commodity units.

"Big Data"

Clustered data stores can handle large amounts of data, but application is more function specific. When loading business transactions, the attributes have to be *mapped* and *transformed*, and there is s huge amount of work to be done here. Some firms are loading ERP data where the mapping can be predetermined across larger client portfolios to make sense, but as transactions migrate to more diverse application portfolios and algorithmic environments, sophisticated sampling and modeling competencies will be essential.

Algorithmic Processes

Algorithmic processes have been managing transactions since the 90's *(wall street runs on algorithmic trading)*. But implementation has exploded as the expansion of machine learning and AI have extended processes into human interface and non-parametric spaces. This creates some concern that are outlined in EY's paper:



Algorithmic explainability. Even where a model offers a high level of accuracy the way in which it achieves this may *not be easily explained given the complex and evolutionary nature of the algorithms used*. Where models are used for things such as credit scoring there is a risk that unless appropriate ways of generating details of how it reaches decisions are included, it may be difficult to justify decisions or demonstrate their correctness if challenged. Guidelines in the new General Data Protection Regulation (GDPR) coming into force in May 2018 cite that, individuals should have the right to obtain an explanation of how a decision based on automated processing was reached.

Algorithmic auditability. As live models constantly evolve, learning from new data, *if auditability is not designed in from the start it will be almost impossible to assure these models and the decisions they make after the event*. There are currently no consistent guidelines in place for what should be done to preserve an audit trail *(e.g. storing interim versions and details of random seeds used in training the model).* •

Algorithmic consensus. For many applications it may be difficult to reach a consistent *(and appropriately justifiable to all stakeholders)* consensus as to what features are relevant.

Algorithmic suitability. There has been a *huge rise in 'democratisation'* of data science with many companies offering off the shelf algorithms that can be quickly deployed rather than these being developed in house. There is a risk that they have not been designed appropriately or that they are used in ways they were not designed for.

Comments

- As we've studied, Machine Learning algorithms "learn" from training data, using *algorithms within algorithms* to find parameter values that optimize fit. Many algorithms are free to create parameters as needed, and often end up a massive number of parameters that only machines understand. We call these non-parametric, they're really "infinite-parametric". EY's point here is that this doesn't leave an explainable audit trail *(we call this "interpretability" in our courses).*

- So how do we audit these algorithms? One approach suggested by EY is to snapshot and map parameter settings *(set.seed)* to reproduce outcomes ("*storing interim versions and details of random seeds used in training the model" to "build auditability in from the start")*.

> *There are challenges with the practice of auditing algorithm internals:*
>
> ***Complexity**. Many automated processes are really complex, executing within large ensemble models that continuously pass parameters. And many run training and operating instances in parallel. Rationalizing logic flow at this level is well beyond the capabilities of assurance teams.*
>
> ***Performance.** While algorithmic processes commonly have alerts for thresholds, every built-in test slows execution and performance is a big concern (standard enterprise benchmarks are 5k/sec transactions). There's going to be reluctance to building audit logic into the algorithms.*

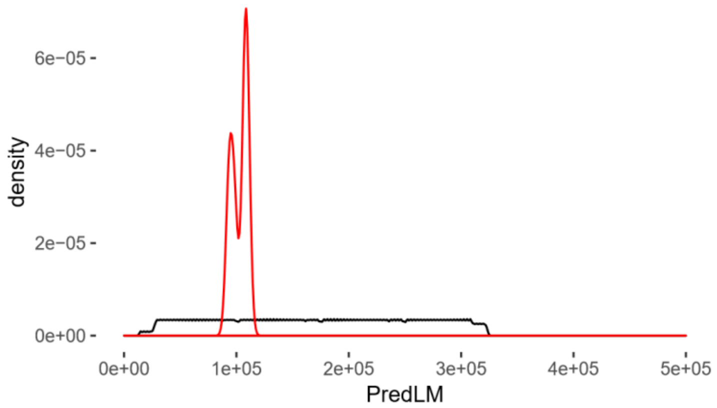
1. EY's point about "a huge rise in 'democratisation' of data science" is an important point. Demand for data science talent is exceeding supply, yet project delivery is driven by a market with i*nflated expectations*. Projects are deploying with algorithms selected and built by marginally competent resources.

BAUER
COLLEGE OF BUSINESS
UNIVERSITY of HOUSTON

Testing Algorithms

Rather than verifying the internals of an algorithm, we can take a higher level approach and focus on algorithm inputs and outputs. For example, we know that, given a distribution of inputs, an algorithm will produce a distribution of outputs *(and different algorithms with different parameters produce different distributions, or "**signatures**")*. So, we can create a grid of all possible inputs *(considering validation control testing)*, and assign probabilities to outputs using a Bayesian model.

The example below is taken from an application project we'll study later. The plot below shows how a decision tree *(red),* produces a very different distribution than a linear model *(blue)*.



We will explore this approach further in the coming lectures and exercises

Reviewing Key Concepts

- The challenges facing Assurance teams in the future are substantial. Transaction environments are becoming higher volume, more complex, and less interpretable.

- Current audit practices are inadequate for addressing these challenges.

- Emerging technologies provide some enablers, but also create many challenges, and the learning curve will cause some pain in the early stages.

- There are many competency gaps in assurance and these gaps are growing. A good career direction!

We have introduced Bayesian modeling as a structured and statistically valid approach to modeling in these environments and explaining transaction behavior. The rest of this section focuses on application of Bayesian modeling to transaction scenarios.

In the next section, we will review the scope and characteristics of corporate transaction environments, and then move into modeling.