WORKING WITH PATTERN COMPUTER

Ty Carlson

CTO

Pattern Computer Inc.
38 Yew Lane
Friday Harbor, WA 98250

References to specific products may be registered trademarks of their respective companies.

Introduction

In the previously published Tech Note, "Introduction to Pattern Computer," we provided an overview of what it is like to work with Pattern Computer Inc. (PCI). In this document, we dive deeper into that topic and address specific details regarding the process, discuss expectations for both our customers and the Pattern Computer team, and provide answers to frequently asked questions. Some parts of this discussion will be familiar from the previous document, which is good for consistency, but much more detail is provided here, with specific examples given where appropriate.

Pattern Computer was formed to discover patterns in high-dimensional datasets; think of it as either a massive table, or a set of tables, wherein the number of **features** (covariates, factors, or simply "columns") and **observations** (samples, events, or simply "rows") can be in the tens of millions or hundreds of millions. Over the last couple of decades, corporations and research institutions have been capturing manufacturing, mining, energy, aviation, business intelligence, and customer data. More recently, sensor, IoT, and customer location data, as well as online "clicks," have been added to the big data lakes residing in their datacenters.

What does it mean to "discover" a pattern? It means that we can identify and rank the most important features associated with a specific outcome. In engineering, for example, we have identified a specific motor-current threshold under a specific test condition which accurately predicts a premature part failure in a critical control system. There were hundreds of different parameters in the test bench, but this particular test (or set of tests) is the strongest predictor of part failure. Under current test scenarios, it passes QA validation. Having this specific piece of information focuses the engineering team on where to start its internal study to reduce the return rate of these critical parts. In biology, it means that we can identify the genes and gene expressions associated with poor survivability in triple-negative breast cancer (out of 24,000 genes).

Why is it hard to discover patterns in large datasets? One must essentially understand all the relevant data and conditions under which all possible combinations could potentially exist. Given the example of triple-negative breast cancer, identifying all the potential 2-way gene combinations in a dataset of 100,000 patients, using eight servers at 3.4GHz with 128 threads each and given 24,000 genes of interest – all 2-way combinations would take only 33 seconds. Now look at all possible 3-way combinations, and it would take three days to compute. All 4-way combinations, and it jumps to 50 years! That is using a brute-force method.

It is not uncommon for us to find 6-way combinations. (It would take 965,000 years to check all combinations using brute-force with today's systems.) While *recognizing* patterns is relatively simple, pattern *discovery* is hard – we are typically looking for a particular set of features (and the range of values of the features) associated with a specific outcome (part failure, cancer, excessive emissions, and the like). We don't know *a priori* how many features will be involved in the pattern (the "dimension"). As the Pattern Discovery Engine™ (PDE) works through the training data, it builds the list of ranked features. The number of features is not predetermined; the data pattern itself determines the natural dimension of the ranked features in each discovered pattern. Some may be 2- or 3-way patterns, while others may be 4- or even 8-way patterns.

Pattern Discovery vs. Pattern Recognition

Many individuals working with convolutional neural networks (CNNs) have stated that they can do pattern discovery with their CNNs. Well, not *exactly*. The CNNs are trained to be able to *recognize* various objects – for this example, let's say images of street signs. They can recognize images of signs for stopping, speed limit, no-passing zones, city limits, etc. But they aren't discovering a pattern. They've been trained to recognize known patterns based on thousands of different images of street signs. They are recognizing something they've already seen – perhaps at a different angle, or upside down, for example, but it is that same sign.

Pattern *discovery*, on the other hand, is just that – *discovering* something you haven't seen before. Neural networks are great tools for recognizing things that have been seen before or for predicting a specific outcome based on significant amounts of previous data (e.g., Is this loan application within the tolerated risk profile?). But they don't discover new patterns in existing data. What, for example, are the patterns behind poor-performing loans? Moreover, what are the critical factors in making those decisions?

How Pattern Discovery Works

The process for pattern discovery is like that of other machine-learning techniques: the main dataset is divided into a training set, a test set, and held-out data. In the case of supervised pattern discovery (most cases), we also specify the Response feature (the outcome that we care about – e.g., broken part, failed test, cancerous outcome). We'll go into the detail of the process of preparing a dataset for ingestion, but the high-level view is that the training set is used by the Pattern Discovery Engine to discover the patterns; the test set to test the model that was produced, which can be checked many times; and the held-out data to do a final check of the model to make sure it didn't overfit the training and test sets.

We train the Pattern Discovery Engine with the training data; in some cases, this may be hundreds of thousands of rows of data (or more). We use acyclic digraphs to build informative structures related to the training dataset. These structures have been highly optimized to allow us to perform the operations on the dataset in memory, thus providing huge performance advantages for our computational purposes. Members of the Pattern Computer team developed their programming talents in the days when it was common to track every byte of data used in the system. That frugal design pattern serves us well when it comes time to do computational work on the dataset – we can generally fit the entire dataset representation into memory, on large compute servers. We have spent over five years developing the algorithms that process the data and perform the pattern-discovery operations. Through this process, we determine the set(s) of features in the dataset that are most frequently associated with the specific outcome. We rank these sets of features as discovered patterns in the dataset and provide that information as results to the customer.

In addition, we build a mathematical model that best represents the ranked set of features mapping to the response variable. Pattern Computer's ability to identify the critical features that map to the specific outcome of interest, as well as our ability to build a mathematical model of that relationship, is a fundamental differentiator from what is commonly referred to as "machine learning" today: the use of neural networks. Neural networks are indeed powerful predictive tools for predicting specific outcomes, but without learning which features of the dataset have important relationships in the outcome, and what that relationship is, there is very little *understanding* of the system and how it works. It's like the high-school–level quest for an "answer" versus collegiate-level work, which is to understand how a system works, how it behaves, so that we understand the nature of the outcome and which inputs have what effects on the outcome. What can I do to effectively change the outcome? Neural networks by themselves won't tell you that and certainly cannot *do* that.

Why Is Knowing the Ranked Set of Features Important?

One of the advantages of pattern discovery within complex datasets is that we can take on hard problems. A couple of examples will help to illustrate the importance of knowing the ranked set of features. One such problem is in the area of human diseases. Our ability to identify the components that affect the outcome (and by exclusion, those that do not) changes the rules when it comes to taking on diseases such as breast cancer. In the breast-cancer example, the dataset we used was METABRIC[1], which was published over 10 years ago. It has information on 2,509 primary breast tumors, with 548 matched normal. It identifies both the ranked mutated genes and a ranked list of the copy-number–altered genes in a list.

Since the publication of METABRIC, there have been no specific gene therapies identified for a subtype of breast cancer called triple-negative breast cancer (TNBC). Why is this hard? Because there are approximately 24,000 genes of interest in the human body. Identifying a single gene to target a cancer is not typically effective, as the human body has a system of pathways in which genes interact and inform one another and will often move to block a therapy (as it is seen as an invader). It is important, then, to understand the system of genes working together in an anomalous manner to support the cancer development.

Finding the pattern of genes working in an anomalous manner is something that pattern discovery is excellent at doing. Using a subset of 1,385 patients, we used the map of 24,000 genes to determine which had anomalous gene expressions for those patients with TNBC versus normal gene expressions. The Pattern Discovery Engine provided a ranked set of patterns showing genes that had anomalous gene expressions. When shown these results, an expert in breast cancer oncology immediately recognized multiple pathways based on the ranked list of gene associations with positive TNBC. He commented that while he had suspected such pathways were involved, this was the first time he had actually seen them presented as a system working together! We have since taken this information to create a gene therapy targeting these genes together; two of these therapies have successfully completed four series of lab testing and are proceeding.

In the case of flight operations, we were able to use the ranked set of features to determine that the overall cause behind flight departure delays in the US was not associated with weather at the departure or destination airport (as might be expected), but rather was due to en route weather – massive storms, which restricted the number of available jet routes along their direction of travel.

[1] Molecular Taxonomy of Breast Cancer International Consortium https://cbioportal.org/study/summary?id=brca_metabric

This was again a case in which the Subject Matter Expert (SME) looked at the ranked list of features and recognized what the pattern indicated. The PDE doesn't know anything specific about the domain in which it is performing pattern discovery. It simply determines which features are most highly associated with a specific outcome.

We've seen similar "Aha!" moments from other customers' SMEs as the ranked list is revealed. They recognize the features and values associated with the event they are looking to understand – e.g., the specific values above or below certain values when specific valves are open or circuits closed – and suddenly they "see" the pattern. Sometimes their reaction is, "Well, *that* shouldn't happen."

The PDE has been referred to as a "hypothesis engine." Perhaps that's true, but it is more than that: it reveals the factors associated with specific outcomes. If you *don't* see a feature in the dataset in the ranked list of features, it also informs you that the feature is not a major contributor. Finally, the PDE identifies which features are true outliers, *playing no role in the specific outcome*. Knowing which are *not* contributing features can be as important as knowing which are.

A Typical Pattern Computer Engagement

What is it like to work with the Pattern Computer team?

Fundamentally, engagements start with discussions about the overall business need and the respective dataset(s) for the business – basically, high-level framing of the project to make sure everyone has the same frame of reference and goals for the project. Those opening discussions assist our team in making sure we have the right members assembled for follow-on discussions, etc. Following that, we have a more detailed discussion on the specific business issue and an introduction to the dataset, including:

1. Size of dataset, tables, rows, columns
2. Scope
3. Nature (and appropriate precision) of the feature set - are the data objects:
    a. Continuous
    b. Enumerated
    c. Binary
    d. Ordinal
    e. Alphanumeric
4. How to handle missing values:
    a. Delete row/column
    b. Impute the value
    c. Average nearest values
    d. Zero
    e. Other
5. Objective:
    a. Supervised (which row/column contains the objective values)
        i. Part failure
        ii. Performance failure
        iii. Late flight departure
    b. Unsupervised

i. Looking for similarities, or natural clusters, within the dataset

Once we understand the size, scope, and nature of the dataset, Pattern Computer proposes the business contract to outline the project, confidentiality, timeline, deliverables, and associated costs. Pattern Computer will also request a point person for the overall project on your team; a Subject Matter Expert (SME), to discuss the nature of the results and ask for specific data clarification; and a Project Lead, to work through iterations of the results set. (The Project Lead and SME may be the same person, at your discretion. With an agreed contract, Pattern Computer will then assign a team to the project.)

Next, the teams will meet to specifically discuss the dataset, at which point Pattern Computer gains access to the dataset based on the agreed terms of the contract and in consideration of the data and custodial requirements of the dataset. The Pattern Computer team will then review the dataset and may run it through some screening tools to understand the nature of the dataset, check for missing values, and identify correlative, or colinear, features to better understand the "signal" in the dataset and which features may represent the most information in the dataset. These findings will be shared with your team and discussed. Once the initial check on the dataset is complete, PCI then runs the dataset through the PDE and checks the result set and related indicators.

In Supervised Mode

The PDE run may discover highly correlative values (substantiated via multiple metrics) and may drop some features from subsequent runs to get a well-defined result set focused on the primary features responsible for the specific outcome. The PDE will produce:

1) A specific list of ranked features associated with the outcome
2) A mathematical model that, using the list of ranked features, accurately models the response against held-out data
3) A detailed mapping of feature/sample interrelationships

Given the information regarding the ranked features, you can load the information into our Dimensional Navigator™ and observe relationships of up to 8 dimensions in virtual reality.

The model

Representing the relationships of the features to the response variable (e.g., the outcome), the model allows for a detailed understanding of the nature of the system being studied. In some cases the model is simple, while in others it can be quite complex – often based on the high variability seen in some output responses. We produce the model in a Microsoft Excel format, as many of our customers use Excel to view their datasets – and this allows them to do a model analysis themselves. Seeing stated accuracy in the Excel tables (when possible, given size limits in Excel) allows the data scientists to do hands-on investigation of the relationships and develop new insights based on these results. We can produce the model in other formats as well, but customers have found this to be very handy for direct analysis of their data.

Updating the model

When new data becomes available, PCI can periodically update the model based on the nature of the business agreement. We can produce one-time results with models, or we can update the dataset with the addition of the new data or application of a sliding window (for example, only the last two years of data) to produce the updated model. Model performance is reported against both training and held-out data at the time of model generation. Models are updated by retraining, and as such, their updated performance and other statistics are reported as each new generation is produced.

In Unsupervised Mode

If there is no specific response variable to model against (i.e., whether the part failed, emissions exceeded limits, etc.), we identify and characterize the clusters of information in the dataset, noting the key features and/or thresholds which prioritize membership in the cluster. By identifying these clusters, you can identify similar features that may stand in, or masquerade, as another feature, thus reducing the impact of an individual feature.

Once the result set is produced and verified by the PCI team, the team will contact you to discuss the initial results with your Project Lead and SME. This is typically a very informative meeting for both you and PCI; your SME will often see a connection among the ranked features and be able to identify a system wherein these features all contribute to the outcome. Not only is the ranked list of features very informative to the SME, but also which features are *not* present, and perhaps which features played no role in determining the specific outcome. It is common for the results to produce multiple patterns – some are completely independent of the other patterns, while others may be specific variations of the first, in which one or two of the features in the primary pattern are not present in the lesser pattern(s).

From this point, the interactions between your data-science team and our discovery team take their own course. In some cases, we may have identified the characteristics of a latent variable, and together we will work to identify the missing feature and obtain the data regarding that feature/subcomponent. It is not unusual for the SME to request the ability to “zoom out” to a broader scope to include other features to see how they may be playing a role in the outcome, perhaps as identified by the first set of results. Alternatively, an investigation into a subcomponent, filtering out information to focus in on specific instances or interactions, may take place once the overall pattern is known.

At this point, your data scientists will be more familiar with the PDE and its capabilities, and the runs shift to focusing on related areas where questions have been previously unresolved and now can be understood and modeled. Some of the more interesting patterns are loaded into the Dimensional Navigator for direct observation of the relationships between features.

Dimensional Navigator

It’s been said that “seeing is believing.” Perhaps that’s true; but more than that, seeing the relationships between features creates the opportunity for *understanding* the nature of those relationships: where the datapoints lie, how tightly or loosely they may be clustered, and whether all of the “failed” parts are clustered together with some features and/or are intermixed with other features. Using the Dimensional Navigator, you have the opportunity to view up to an 8-dimensional relationship using virtual reality. You can virtually “fly” through the dataset, change the primary and secondary axes features to see the relationships in different layouts, and investigate individual datapoints and their subtended values directly.

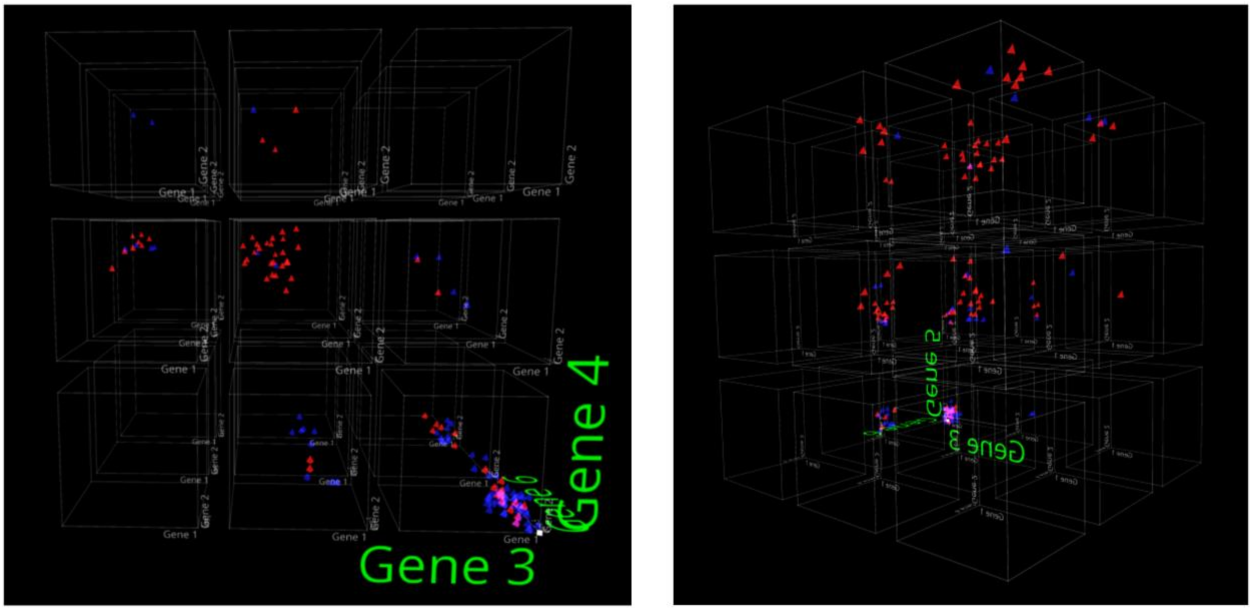
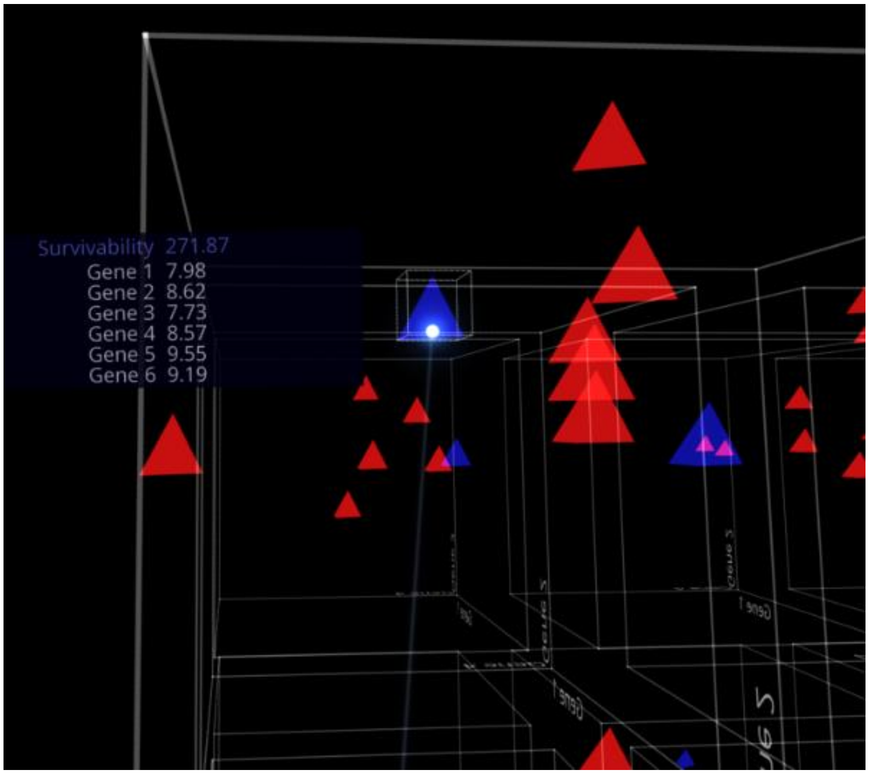


Figure 1: 8-D views using the Dimensional Navigator



Survivability 271.87
Gene 1 7.98
Gene 2 8.62
Gene 3 7.73
Gene 4 8.57
Gene 5 9.55
Gene 6 9.19

Figure 2: Querying an individual datapoint within the 8-D context

There are other options, such as filtering out certain conditions or changing thresholds, as well as focusing on the high vs. low responses to simplify the clusters and understand the nature of the distribution of the responses. Sometimes it's just useful to look at the relationships of the features identified in the pattern to get a sense of the data itself, as represented by the ranked features.

Deliverables

Pattern Computer can provide:

- The preparatory application Sextant, which can be run on your dataset to describe the dataset to PCI without sharing any of the actual data. Sextant outputs a readable JSON file that informs PCI of the dimensions of the dataset and the nature of the data (number of rows/columns of alphanumeric, binary, integer, floating point/precision, strings, image, etc.). Sextant also looks for initial correlations in the dataset, as well as colinear features. Sextant can also check for signal in the dataset. This file can be read and approved by the customer for sharing with PCI. This is not a required step, but it is informational.
- The Results set, which includes:
  - Supervised Mode:
    - ranked list of features associated with the specific outcome
    - mathematical model based on the ranked features which accurately describes the relationship with the outcome; this can be verified with the held-out data
    - ranked set of secondary patterns associated with the specific outcome
    - Each pattern details:
      - the list of observations that support the specific pattern
      - the list of observations that are outliers to the pattern
  - Unsupervised Mode:
    - ranked list of clusters of similar features
    - optimized number of clusters based on overall "distance" between clusters
- Dimensional Navigator
  - This is a Pattern Computer–developed application that takes the ranked features list and presents the representative data in an 8-dimensional virtual space. The Dimensional Navigator uses the Vive[2] VR system and the Steam[3] engine.
- Report-out
  - PCI creates summary reports of the runs, documenting the ranked features, ranked patterns, and overall mathematical model. This includes the input of the customer team (project team, SME, etc.) as the pattern-discovery process develops and is interpreted by the SME into understanding of the system at work and the nature of the relationships and/or interactions. Charts and graphs are often included showing the held-out data and the mathematical model results, in addition to the accuracy values. Additional observations made by the working team are captured and reported-out as well.

The customer provides:

- Project Lead (the point person we will be working with)

[2] https://www.vive.com

[3] https://www.steampowered.com

- Subject Matter Expert (SME) for the project
    - The Project Lead and SME could be the same person
- Prepared dataset(s)
    - Not generally limited by size (if over hundreds of millions, let us know)
    - Ideally, dataset is scrubbed:
        - no missing values
        - no corrupt data
        - type consistent
    - If data is missing, what are the rules?
        - delete the observation
        - impute the value (by…)
        - zero
        - NaN (not a number)
        - n/a
- If supervised, provide the Response feature:
    - What outcome are we modeling?
        - e.g., part failure, manufacturing delays, test failure

Pattern Discovery

What makes PCI's approach to pattern discovery fundamentally different?

Our proprietary technology is orthogonal, yet highly complementary, to contemporary methodologies. Our methods use techniques drawn from nonparametric statistics, time-series analysis, clustering, deep neural networks, topic modeling, reinforcement learning, and adaptive/learning Markov decision processes.

The uniqueness that PCI offers centers on our ability to interrogate fitted models – to gain insight into the functions learned by state-of-the-art AI and ML algorithms (including our proprietary algorithms). PCI's algorithms learn and identify low-dimensional surfaces and response functions (explicitly functionally specified). Such representations correspond directly to testable hypotheses – potential discoveries – and enable counterfactual reasoning.

For purely predictive settings, wherein the cost of being wrong is low and the user requires little or no insight into the problem, PCI and other vendors (i.e., Palantir/Google/SparkCognition/Microsoft, etc.) are on more-or-less equal footing. When the cost of being wrong is high, or when the user is required to learn new engineering, control, or physical principles, PCI offerings are more strongly differentiated.

A PCI analysis run starts as they usually do: high-dimensional, heterogenous (an allowable mix of binary/categorical/nominal, count, ordinal, interval, and ratio), minimally curated data is processed to efficiently and quickly determine "locally" nonlinear functions of "interaction sets" of variables ("patterns"). The term "locally" is important here, because PCI learns the notion of locality, or metric embeddings, directly from the data – we never impose or assume metric or topological principles. This stands in stark contrast to topological data analysis (TDA), as well as visualization techniques such as

UMAP, t-SNE, Spectral Eigenmaps, and the like – all of which require the user, at some stage in the algorithm, to assume a metric embedding for the data.

After we've learned an embedding, our approach allows for the learning and identification of *p*-th order interactions (e.g., p = 6) among input heterogeneous variables even if no lower *r*-th order interactions exist among the interaction set (e.g., even if r = 1,2,3,4,5 interaction subsets do *not* exist). Hence, we can identify non-additive interactions that violate the principle of marginality, in contrast to forward procedures – e.g., H-statistics. Further, we never assume the form of an interaction; we learn functions and their parameterization and present these to the user as potential discoveries – patterns, hypotheses.

As others do, we rank and assess our confidence in discovered patterns in a causal inference framework. While there is some "secret sauce" in the details, we are broadly borrowing from the causal inference literature at this stage. The ranking aids in decision support and the prioritization of studies or subsequent tests.

The learned response functions are compositions of per–interaction-set functions, so one can clearly see how/why the interaction-set variables functionally interact and thereby provide explanatory power to the user. At a higher level, the interactions between the interaction-set functions give a "zoomed-out" view of how clusters of variables interact, and we can move up this hierarchy to obtain insightful and usable levels of abstractions. In this way, the interaction sets form equivalence classes of lower-level variables that allow a hierarchical understanding of the behavior of a system or process. This procedure effectively learns the manifold structure of the input-features-to-response-variable manifold and easily allows for the identification of irrelevant variables and a hierarchy of variable relevancies.

We have applied our technology to several application domains to identify known patterns (an important proof of principle) and, most important, to discover novel patterns that are acknowledged and accepted as such by domain experts. Using our proprietary combination of hardware and software, we can do this quickly and at scale.

Flexible Engagement Model

Pattern Computer can engage on a specific project or dataset or in a subscription model, whereby we will work with you on one or more datasets to partner with and train your data-science team on the use of the PDE and how to:

- interpret the relationships between the ranked features
- map the identified patterns to the associated observations
- identify true "outlier" features
- check for latent variables (features)
- "zoom in/zoom out" on the dataset
- use hyperparameters
- apply tips, tricks, and insights in using the toolset

Pattern Discovery as a Service

Pattern Discovery as a Service (PDaaS) is currently in development, along with the staging of the high-performance servers. Once the customer's data-science team is familiar with the process of using the Pattern Discovery Engine, the use of the PDaaS will be fairly straightforward – not only the initial runs, but also the follow-on runs, which often zoom in and zoom out on the dataset to reveal more insight into the nature of the pattern and the specific contributions and interactions of the feature set.

Cloud-Based Solutions

Pattern Computer has its own Amazon Web Services (AWS) instance available to scale our PDE runs to meet specific deadlines or in cases where we desire to do a broad series of runs with variations of the dataset(s). We have the same capability to perform runs in the cloud as we do in our datacenter. To be able to comply with custodial data requirements, we can also configure our software to run in a subtenancy within a customer's AWS tenant, such that the data never leaves the customer's AWS instance. The subtenancy is ultimately managed by the customer. We ask for access to the metering data so we can aggressively track costs data to make sure that it matches our expectations. Within the down-level tenant, we manage access to our executable files and audit access.

Who pays for the cloud-based costs?

Payment may be a matter for negotiation. We have been recognized by our customers by how aggressively we controlled costs in the AWS cloud. Our algorithms are highly distributed, so we can run efficiently within a cluster of low-cost servers or run on larger, scalable boxes, depending on the driver (time or cost). We also carefully manage storage costs, which can otherwise add up quickly. Ultimately, with longer-term engagements, we have moved to placing our own server within the customer datacenter. This is because the storage required for the larger runs, and the ability to truly have full access to the machine hardware (and where we know the exact specs), became an important advantage. This provides overall cost savings, as well as flexibility to perform multiple investigative runs with variations of the dataset, without the AWS "meter running."

A Customer Engagement Example

An interesting problem set was brought to us by a company that manufactures critical control systems. The company was seeing warranty claims increasing on a specific part on a specific device. This was clearly a significant issue for the manufacturer. Additionally, unscheduled downtime for the part incurred very high costs for its customers and risked damaging the company's reputation. This was an urgent issue.

The total number of specific parts was in the low hundreds of devices, which can make isolating the problem difficult due to limited data. We worked with the company to identify what information was known about the history of the parts – those that had served their full useful life and those that had warranty claims. We were provided with the QA test information for the parts, based on serial numbers – with the test data recorded by hand on technician-stamped paper records.

Working together, we were able to get the QA information digitized, and we included the test information as well as the metadata, such as test date, test equipment, test engineer, and inspector information. Overall, it was not a large amount of information. We had the data for testing the extension and retraction, distances, times, current, and power requirements of the critical control systems for completing the test tasks. In addition, there was a recorded graph of the rate of extension and (overlaid) retraction. The data, though, generally did not include the nature of the failure. Many of the warranty claims simply noted “failed part, replaced.” Given that we didn’t have much in the way of specific information on when or how the failure occurred, we would have to use the initial manufacturing test data to see if it could identify a pattern behind the failed parts.

Running the data through the PDE, we saw a pattern emerge of the parts that had passed the QA tests but had been returned within a three-year window for warranty claims. In one of the certification tests conducted in a known test harness, there is a measurement of the motor current required to operate the part. The PDE was able to identify that if the part exceeded a specific amount of current, that part had a very high percentage of premature failure. Now knowing where to start looking, we mapped out the motor-current values for the various (sequential) serial-number parts over time and observed specific patterns of increasing motor currents mapping to failed parts. We were able to identify key windows of dates wherein the parts were very likely to fail, and we suggested looking at the various parts vendors that had changed during these specific time windows.

Also very useful for the company was the discovery that the failures were not specific to an individual technician, inspector, or test device. There was real value in the results from the PDE, not only for which features appeared in the ranked list, but also for which features did not (e.g., individual technicians or test devices). Notably, devices were identified by the patterns that were currently in service which were flagged for early failure – giving the company the opportunity to proactively replace that part currently in service.

Without divulging specifics, the findings were very interesting to the customer’s engineering team, as all the parts had technically “passed” the QA process – but it was clear that during specific periods of time, the amount of *power* required to complete a specific test was a specific indicator of future failure. Thinking through the problem, that is a reasonable association for a potential failure. The PDE was able to identify which of the battery of tests was the most significant, the specific measurements, and the threshold value for indicating the association with early failure.

As a result, the customer’s engineering team is re-evaluating their requirements for “passing” that specific test, while at the same time investigating the suppliers and any design changes on the assembly. The team’s feedback was that they were impressed that we were able to isolate a specific indicator of failure. Once that was known, when graphically mapping the motor-current test values for the various serial-numbered parts (and noting which ones had failed), the pattern was very clear. Now the customer’s engineering team knows where to focus their efforts on understanding the reasons for unexpected resistance in that configuration.

This work has led to future projects with that customer.

Engagement with the Customer Team

During this investigation, we had three one-hour meetings with the Project Lead, two one-hour meetings with the Project Lead and the SME, and a final report-out meeting. Additional time was required of the SME to locate the cutaway drawings and to look for the design change and vendor supplier information. We received the vendor supplier information at midnight on the night preceding the next morning's scheduled report-out. We then folded that information into the final report-out, which aligned with one of the failure sequences. The specific steps were:

1) Discussed the scope/overview of the initial datasets and agreed to digitize the information. We chose not to digitize the Extend-Retract Distance Over Time graph, as it was viewed more as a qualitative piece of data.
2) After the PCI team had reviewed the information, we wanted to understand how the specific control system worked (parts of it were redundant). We requested cut-away and component drawings to assist us in understanding the reference to specific components and how the mechanism worked. By chance, our staff includes an applied physicist, control systems experts, and two persons with specific experience in the customer's domain; it was useful and informative to have access to their additional insights.
3) Follow-up call with the Project Lead and SME. We received the images we had requested, and the SME explained the operation of the part. We had made an initial run with the PDE and had some questions about the specific motor current. We also had stated in the technical notes that there was the equivalent of a "recall" on specific serial numbers. The description of the change didn't make sense to us. We asked for more information, as we wanted to know how that might be related to the motor-current pattern we were starting to see. The SME explained the purpose of the recall.
4) Follow-up call with the Project Lead. We let the Project Lead know that we had discovered a pattern and that we had some initial graphs showing an association with the failed parts. We requested information regarding any vendor changes (reasons for different parts being slip-streamed) and/or design changes (including dates and the impacted serial numbers).
5) Initial report-out to the Project Lead and SME of our findings. They were interested and pleased with the results. We had a scheduled date for the report-out, but the SME requested the preliminary deck immediately to discuss the findings with his team. He particularly wanted the graph highlighting the motor currents versus serial numbers and failed parts, which basically captured the issue in a single image.

The final report-out was to the Project Sponsor (VP Engr) along with the Project Lead, the SME, and members of his team. The final report-out meeting was followed up with a formal report to the customer, including all the pertinent information, charts, tables, and models.

Specific Issues or Questions

Custodial Requirements of the Dataset

Pattern Computer recognized from the outset that some customers would have dataset regulatory requirements regarding the custodianship of the datasets (often in cases where their business is based on their data). Some business partners have standing rules forbidding any removal of their data from their premises. In the cloud computing world, that typically translates to their private cloud/tenancy in Azure, AWS, or Google Cloud. Hospitals, medical research, and patient data is typically restricted by (at minimum) HIPAA regulations. Research consortia also often have custodial restrictions that the shared dataset needs to be retained in the custodial control of their respective members.

As a result of that insight, Pattern Computer's PDE was designed to be able to operate outside of PCI's datacenter resources. At present, Pattern Computer has an AWS implementation that we can run inside our private tenant or in the customer's tenant. In cases where we run in the customer's tenant, PCI and the customer both agree to have very restricted access to the Pattern Computer subtenant, with separate access for administration and auditing. The Pattern Discovery Engine must also be provided with access to a specific restricted resource with the sole purpose of authorizing and auditing runs within that subtenant.

In cases where the dataset must reside on the customer's premises, Pattern Computer can arrange a custom engagement wherein one of our computers can reside in the customer's datacenter with strict administrative and auditing control and with highly restricted access on both sides. The PCI computer includes a hardware acceleration card that must be present for computing offload and security controls. The computer must be allowed the ability to "phone home" for authorization and auditing of runs, but no specific information related to the data or contents of the data is shared. The information simply references a job ID, the details of which are stored on the customer's premises.

Customer Resource Requirements

The customer resource requirements depend on the size and scope of the project. For example, how many subsystems are involved, and how many teams are involved? We work to be efficient with our time as well as our customer's. We focus on "doing our homework" and reading all pertinent information, so any of our questions and asks are targeted, specific, and clear. Once we understand the scope, we can be much more specific as to the requested resources. We make effective use of email for communications and have meetings when a virtual or face-to-face meeting will be the most effective means of communication.

Nonstandard Information Controls

We have been engaged by customers with nonstandard information controls. When taking on such projects, we need to understand what it would take to be compliant with the nonstandard information controls or any specific exclusions that may impact other existing and/or future projects. If the terms are acceptable, we will work to get the appropriate team member(s) compliant. For example, we have had several of our team members obtain specific certifications to be able to access genomic data for humans. There may be other options as well, such as PCI exposing a web service interface to the PDE locally within the customer network. Another option may be to write up a specific agreement.

On-Premises Requirements

Projects in which no data, results, models, or any interim information can leave the customer network are not problematic for us. We need to understand the restrictions and work with your datacenter IT staff regarding placing a server in your datacenter. We'll also need to understand the necessary requirements/restrictions from both companies regarding the access control, allowed hardware and software configurations, monitoring, and auditing capabilities. There are generally two options:

1) PCI to provide a PCI server into your datacenter. This would be server-based on the PCI internal specification with configuration optimizations for the Pattern Discovery Engine. This includes the FPGA accelerator card we use for acceleration of our algorithms. PCI would need remote access to the server, but all the data and all the runs occur locally on that server within the customer premises.

2) PCI to purchase a server in the customer fleet in the HPC class. In this case, the server is compliant with the customer's IT department base build. PCI would need to verify that our software configuration is compatible with the base operating system, and so on. We would need to add the FPGA accelerator to host the PDE there. It would need to have very restricted access and auditing turned on with separate permissions for admin and security admin (so admin can't delete the auditing log).

If the data needs to remain on-premises at the customer site, PCI will need secure access to our secure server with our FPGA card installed. This is a model that we are familiar with; in such cases, our chief security officer will work with your team regarding specifics.

How can Pattern Computer create the model if the data can't leave the customer network?

The modeling is performed as part of the Pattern Discovery process. When the PDE identifies the ranked list of features, we then discover the patterns and build the overall model. This would require either the presence of a PCI server secured within the customer network compliant with the customer's data policies or PCI purchasing an HPC-class server compliant with the customer's datacenter fleet, as well as adding an off-the-shelf FPGA accelerator card.

Complying with program-specific controls

We would need to understand what it would take to be compliant with the nonstandard information controls and work to get a member of our team compliant. As stated above, we have had several of our team members obtain certifications to be able to access genomic data for humans. There may be other options as well, such as PCI exposing a web service interface to the PDE locally within the customer network. Another option may be to write up a specific agreement.

If the PCI server is on the customer's premises, what does the customer need to support?

In our experience of on-premises configurations, we had a situation in which the new customer-compliant "fleet" server (which was the same spec as many of the other servers) would hard-crash under heavy load. We asked the admin to enable detailed debugging so we could verify with the OEM that it was the CPU that was at fault (it was), and the new machine was swapped out under warranty. The only other request we have had in approximately two years is that we added three 16TB hard drives to scale to the requirements of the dataset and our intermediate checkpoint files.

We need external (SSH) login to the on-premises server to initiate and supervise runs performed on that machine and to update our software as needed.

The on-premises PCI server needs outbound access to PCI for license validation, as well as inbound access from PCI for supervision and maintenance. Customer IT admin access should be limited to a small number of identified individuals with a documented need for access (security, etc.). Hardware and software audit logging must always run.

Is Pattern Computer integration-compliant with US Department of Defense contracts?

Pattern Computer does not currently perform on any Department of Defense contracts and has not sought certifications (such as CMMC). We operate in compliance with DoD, NIST, and other guidelines, however, so achieving a given formal certification should not be difficult if the need arises.

Access Control

Overall, we limit access on a specific need-to-know basis. We are there to work with the data and discover the patterns as efficiently and effectively as possible. Anyone not actively assigned to the project should not have access to the server and/or dataset. We respect the need for security and limited access. We understand the strict need for confidentiality and demand that of all our team members. Customers may limit individuals in Pattern Computer who are involved with a given project to a specific access list.

Control options

PCI uses a four-level control model for access and handling of sensitive but unclassified data. (PCI does not generate or process classified data.) Employees are regularly trained on sensitivity levels, how to label data, and what types of devices or storage can be used for each level. PCI enforces a mandatory access control model for files and data stored in our development systems and datacenter.

Security clearances

Currently, one Pattern Computer employee holds an active US (DoD) security clearance.

Latent Variables

Latent variables are those variables that aren't seen but which may be impacting your results. The PDE can look for latent variables and flag that possibility. We look for situations in which the introduction of a different factor being present or not would further improve the accuracy of the model.

One such example is illustrated by the most important features identified in the Flight Operations dataset described earlier. The purpose of this dataset was to understand the patterns behind flight departure delays at US airports. Using the US Department of Transportation's Bureau of Statistics record of 7.2 million flights during calendar year 2018, plus aircraft and logistical information from the Federal Aviation Administration and local weather information for the departure and arrival airports, our initial pattern-discovery run yielded:

FLIGHT DATE, TAIL NUMBER, ROUTE

That seemed unusual to our internal SME, as he had flown commercially as a passenger for 12 years when he worked at Microsoft. He expected to see things related to departure airport weather or local tower delays, volume delays, etc. He was interested in why flight date would play such an important role, so he looked at the dataset related to this pattern. The data indicated that there were flight departure delays on specific days (FLIGHT DATE) which impacted flights crossing specific parts of the country (ROUTE), and that specific planes (TAIL NUMBER) were further impacted because they had regional routes where they had to transit these areas multiple times a day.

When the SME looked up those dates in that region of the country, he found reports of massive regional storms covering hundreds of miles. The dataset had included "origin airport weather" and "destination airport weather," but had not included any "en route weather." What our SME suspected was that the en route weather was closing the available jet routes crossing the path of the massive storm fronts. Based on this insight, he added the FAA Operational Network (OPSNET) information, which includes flight delays initiated as part of the FAA's Traffic Management Initiatives (TMIs) and added TMI delays due to weather (initiated by either the origin [TOWER_FROM] or destination [TOWER TO] airport). After adding the OPSNET delays to the dataset, the top ranked pattern was:

TOWER_TO, SCHED_DEP_TIME, LEG, TOWER_FROM.

– which more accurately captures the flight departure delays as initiated by the FAA OPSNET.

The Pattern Discovery Engine can automatically look for the existence of latent variables in the dataset, but this is a good example of how a latent variable may crop up through the pattern-discovery process. When we do identify a latent variable, we can verify that is the case, as the associated increase in predictive utility of generated models gained by the inclusion of the latent variable can be measured.

Scalability

Pattern Computer's modeling capabilities are highly scalable, as they are pure multivariate equations. Such representations are highly optimizable by current equation interpretation technologies, such as Microsoft Excel. Moreover, since the generated models are stateless equations, they are highly parallelizable at the implementation level.

Data Types

Are there restrictions on the types of data that PCI's techniques can work with?

While much of our work to date has been in the tabular data space, we also have expertise in imaging and other data formats, such as CAD and NLP. This considered, hybrid approaches are also possible – for instance, images may be classified and labeled using industry standard techniques, and those classifications can serve as input to further processing stages as part of a larger tabular whole. These customized approaches to data ingestion are highly specialized and problem-domain–specific, and would require deeper specific analysis to address. However, they are within the scope of our core technologies and within the areas of expertise of our team to address, should it become desirable to do so.

Our primary algorithms and technologies are particularly well suited to tabular data, but this does not mean that we are strictly limited to such data. More complex data relationships may be handled with a hybrid approach on a problem-specific basis, should the need arise.

What file formats can be ingested by PCI?

Datasets in CSV form, or in tables that can be easily represented by CSV form, are the mostly easily integrated with our engine. We can also accept data in HDF5 format. If the data provided is hierarchical in nature, we would need to have the schema (if not built-in) to understand that data. This said, our data access object (DAO) layer is sophisticated and flexible. We also have specialized imaging-recognition systems in development that can read standard image formats (JPG, IMG, and PNG).

High-Dimensional Data

What data requirements need to be in place for PCI's dimensionality reduction techniques to be successful?

First and foremost, the input dataset should be as clean as possible, regardless of its overall dimensionality. High dimensionality implies large dataset size, and scrubbing such large datasets prior to processing for analysis and model generation can be a computationally extensive task. Moreover, such scrubbing can be highly sensitive to subtle issues that only an SME might fully appreciate.

While our engine can gracefully handle missing values, these should be kept to a minimum. As many of our algorithms are supervised, an accurate response variable for each sample is also important. Our algorithms allow for the tracking of which observations inform various stages of the analysis and modeling phases of our engine, and as such, assigning unique identifiers to each observation also serves to improve the overall end-to-end process, as it allows for deeper investigation and auditing.

Once data has been cleanly ingested into our engine, measures of model accuracy and other weighted ranking metrics are in place to assist with the process of determining what the data has to say, regardless of its dimensionality at the onset. As every hypothesis can be accompanied by an audit trail back to specific observations that contributed to its formulation, measures are in place to mitigate the general limitations of working in such high dimensions.

It is worth noting that the PDE is well suited to datasets with a high-signal continuous or binary-response variable to train on. As already mentioned, dataset hygiene is particularly important to favoring successful analysis and modeling outcomes.

How does PCI's approach to high-dimensional datasets differ from the large body of existing research in this area?

Rather than casting a high-dimensional dataset as something to be *reduced* to a smaller set of dimensions, we approach the problem to one of dimension *selection*. In certain contexts, feature subsets impact outcomes of that context; in other contexts, different feature subsets apply. For instance, which features apply to eventual part failure may not have any impact on other (nonetheless interesting) outcomes that might be a separate topic of investigation. Our approach is to find those subsets of features that propose to most inform outcomes on certain subsets of the observations, and then to test these proposed relationships (hypotheses) by building predictive models in those spaces. Global models are then developed from the most performant local models. The principle of parsimony naturally yields models that are greatly reduced in their dimensionality, as not every feature has impact in every context or subpopulation – only as many features as are required to predict outcomes in a way that is as generalizable as can be achieved, but no more.

Sometimes our initial investigation might suggest that our original dataset dimensionality is too small (no matter how large it may at first appear by feature count alone). By increasing the number of features, we may then find the truly underlying patterns in the overall set. We let the data speak for itself through various measures – some proprietary and some well-known in the field.

This approach is outcome-oriented and tied to the semantic space of the features as they relate to the observations those features measure. In this respect, dimensionality reduction is a necessary (and extremely useful) byproduct of pattern discovery, rather than an end in itself. At no time are features blended or embedded into opaque abstractions; the features of the final model are in the scale and units of the features as measured on the observations. Projection into the response space is accommodated by the format of the model itself, and as such, the interpretability of the model is as close to the original semantics of the dataset as possible. What the SME brought prior to a run will likely apply after a run – but now with far greater attention brought to the key impactors.

This approach has yielded some impressive results. Where others might have been able to "reduce" covariate set size to 1.2M from 39M, we have achieved better predictive results on only 167 covariates of that same dataset. We have achieved this not by blending and eliminating features or reducing their number, but by algorithmically identifying interactions, relationships, and patterns in the most informative (and naturally much smaller) subsets thereof.

Data Requirements

How much data is required for PCI's techniques to be successful?

Data quantity requirements have varied by project. In some cases, we have been able to produce faithful, parsimonious models with very few samples. In others, we have examined over 50,000 samples. Smaller training-set sizes and the methods used to gather and select the samples will limit the nature of the conclusions one can draw about the generalizability of any results, regardless of their reported accuracy. This may be an important consideration when training on small training sets.

How does performance scale with the data?

Our pattern-discovery and modeling algorithms were specifically engineered to scale to very large datasets. Such performance and scaling considerations were never an afterthought. As such, from our DAO layer upward, the Pattern Discovery Engine is designed to deal with high-data–size requirements. How this might map to a specific project is subject to the actual relationships of the features in datasets to be processed.

Summary

This document provides an overview and working example of what it is like to engage with Pattern Computer Inc. to discover patterns in your datasets. While the Pattern Discovery Engine is dataset- and domain-agnostic, the partnership between Pattern Computer and our customers is unique and specific to each customer. We focus on the datasets and listen to the SMEs describe and detail the datasets and the meanings of the various features and how they are related. It's very common for the datasets to have exceptions that we need to work through, or the like.

The working relationship and the engagement between the Pattern Computer engineering team and our customers is highly valued and leads to a creative dynamic to understand the results and create additional value and insights for the customer. As our customers' SMEs understand the discovery process, they use the tool's ability to zoom in and zoom out to understand the full scope of the patterns, as well as identify and understand the distinct patterns within the dataset.

It truly is the difference between just having an answer and understanding the system.