**1. Introduction**

The discovery of patterns is an often elusive goal of data analysis. Here we build off of an existing framework for uncovering them called t-pattern detection. In this method, we consider data consisting of some events, each with a type describing it, happening at a specific time over a specified timeline, or observation period. A t-pattern consists of observing two or more of these types of events separated by relatively similar amounts of time more often than we would expect by chance. We search for an algorithmic way of identifying these patterns of events across time because it has been demonstrated that humans can be easily overwhelmed when a timeline is crowded with many events happening, and not see a pattern that would be obvious if only all extraneous data were removed.

The applications of t-pattern discovery have largely been limited thus far to very specific situations. It has been applied to small-scale experiments of human or animal behavior, where their physical movements are monitored by video and encoded manually into the necessary timelines. These have included movements of players in a soccer match, verbal and nonverbal cues of teachers in front of a classroom, and facial expressions exhibited while lying. Each of these shares certain commonalities: the number of subjects is small, the period under observation is relatively short and continuous with *something* happening at almost every moment, and the types of events observed are carefully chosen beforehand so they will be observed frequently and by all subjects.

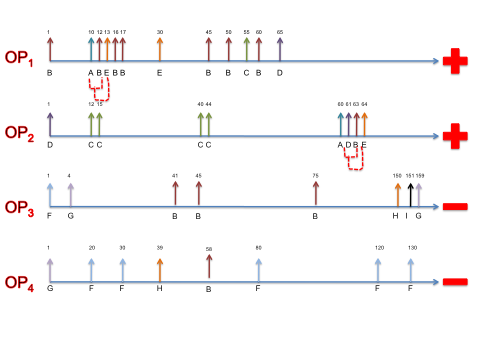
The use of t-pattern discovery in these settings tend to yield large numbers of t-patterns, all of which do in fact statistically significantly deviate from the null model’s method of specifying what we would expect from random chance. However, many if not most of these patterns may not hold any real meaning, and their presence obviates that of the meaningful ones. Thus, analysis of t-patterns is limited to very high-level discussion, such as the number of t-patterns in an observation period, the largest number of events in one t-pattern, or the average number of events in all t-patterns. To compare different observation periods, comparing these statistics limits discussion to whether they differ in how structured their behavior is, excluding discussion of specific t-patterns.

Some efforts have been made to find particular t-patterns that occur more often in one group of observation periods than another, which we intend to build on for the purposes of applying t-pattern discovery to a new setting. In many business intelligence situations, we have information on many customers and would like to predict something about them. For example, imagine we are a credit card company and have records of each customer’s buying history and then whether he paid his bill on time or not. We wonder, can we find individual temporal patterns in our data that will help us predict whether a new customer will pay his bill on time? If so, maybe we can intervene when we see a certain behavior and cut him off before our future losses pile up.

In the existing methodology, the analysis of the results of t-pattern discovery is conducted manually. Clearly, in our classification task, this is not feasible. We propose a technique for algorithmically focusing t-pattern discovery for its downstream use in classification. Along the way, we encounter and propose solutions to problems caused by the inherent differences in the structure of small observational experimental data and large business intelligence data.

**2. Motivating Example**

For illustration, let us consider a motivating example:



Imagine we have 4 individuals for whom we have information about their behavior over a specific period of time. We also have the classes that these individuals fall into that we will later try to predict. Can we find patterns in the aggregated behavior of these 4 individuals that helps us predict which category a newly observed fifth individual will fall in if we know his behavior? In this example, the pattern of an occurrence of A followed by an occurrence of B after 2-3 time units and then a C 1 time unit later was generated in every positive observation period and no negative ones, with the rest of the events being generated randomly. We would like to find this pattern while not being distracted by other patterns that very well might appear in the data but not help us categorize the observation periods.

To do this, let’s organize our data into 4 independent observation periods {OP1, OP2, OP3, OP4}, each of which has a timeline separated into discrete time units. Along these timelines we record an individual’s behavior by marking what types of events happen and the time at which it happens. Our motivating example has 9 types of events {A, B, C, D, E, F, G, H, I}, and we can refer to an instance of event A occurring at time 10 in observation period 1 as A1,10.

With this setup, we use an existing t-pattern discovery algorithm can help us achieve much of our goal. In the next section, we discuss the process by which we hierarchically build these t-patterns by considering one observation period at a time. However, in the following sections, we discuss difficulties that arise when we try to use this process for our prediction goal on realistic data sets:

* How do we incorporate information from across the multiple observation periods when the underlying processes that generated each one might have been very different?
  + Our null hypothesis should reflect that Event B is very common in OP1 but not in OP2 and that Event C might look uncommon in the data set as a whole, but in OP2 it’s actually quite common.
* What happens if we look for patterns in one magnitude of time when an observation period really happens over another magnitude?
  + If t-patterns only have a business use if they are detected on an hourly level (e.g. it doesn’t do us any good to only know the day or week when events happen), but our timeline is years long, our null hypothesis is violated
* What heuristics can we use to prune the search space to make computation feasible when we have large, real-world data sets (both many observation periods and many types of events)?
* Once we have uncovered t-patterns, how do we build a model to classify new observation periods?

With an abundance of data, finding statistical oddities, or patterns, is not necessarily useful. At the end, we not only hope to have found structure in messy data, but we hope to have found only the aspects of the structure that are informative for a decision-making task.

**3. T-Patterns in One Observation Period**

In this section we briefly summarize the statistical framework for discovering a t-pattern within one observation period. Each t-pattern consists of two events, a left and right, and a statistically invariant number of time steps, called a critical interval, separating them. Once we find a t-pattern, it is considered as a new, more complex event type, and thus can be used as either the left or right event in a higher-order t-pattern. In this bottom-up method, we can form arbitrarily complex t-patterns.

We start by considering, one-by-one, all possible t-patterns of length 2, that is, every possible permutation of event types, with the current one under consideration termed the current candidate pattern. Using our motivating example, let’s consider the candidate t-pattern TPA-B­, where the left event (TPL) is and the right event (TPR) is . For each occurrence of , the time until the first subsequent occurrence of , called an interval, is recorded. The distribution of all of the intervals for candidate t-pattern TPA-B­ is used to generate candidate critical intervals , starting with the interval and then moving closer in until all candidates of form [d1, *x*] are considered, when we move d1 up and repeat until all possible pairs are considered.



The null hypothesis assumes that and are independent and that the occurrences of are uniformly distributed over the entire observation period, so if an occurrence of happens at time , the interval should be no more likely to have a occurrence in it than any other interval of length .



Given the assumptions, the probability that occurs at any individual time unit for an observation period is simply the total number of occurrences of divided by the total length of the observation period:



Thus, the probability that any particular time unit doesn't include an occurrence of is:



It then follows that the probability that an interval of length doesn't include any occurrences of is:



Finally, the probability that an interval of length does include one or more occurrences of is then the complement of the probability of zero occurrences:



Using this conclusion, if there are occurrences of for the observation period, we can model the number of them that have an occurrence of in the critical interval after them with a binomial distribution. Here, there are trials, each with probability of success . We would expect of the intervals to include an occurrence of simply by chance. If the actual number of them that include such an occurrence is , we can calculate the corresponding p-value according to the binomial distribution, which represents how unlikely it would be to observe this if our null hypothesis is true. If this p-value is sufficiently small, we say that forms a t-pattern with critical interval. If not, we move to the next candidate critical interval from the values in the distribution of intervals for . This continues until a calculated p-value is statistically significant or all intervals are explored, when we would conclude there is no t-pattern between and move on to the next candidate t-pattern.



If there is a t-pattern found, this then becomes another event type itself, of length 2. We generate further candidate t-patterns to check using this new pattern, e.g. , and so forth. In this way we proceed, building arbitrarily large t-patterns, until all candidate t-patterns have been checked. Because we eventually consider patterns consisting of many events as one event itself, we will frequently refer to the left half of a t-pattern as *TPL* and the right half *TPR*, where each half may be a single event type or a previously discovered t-pattern.



The final step, called the completeness competition, consists of checking for complete t-patterns amongst those found. A t-pattern is considered complete if it is not itself a sub-component of another t-pattern nor does another t-pattern have all of the event types the one under consideration has and more. For example, if is a t-pattern, then neither nor would be considered complete.



**4. T-patterns in multiple observation periods**

**4.1 Proposed Method**

With this definition of a t-pattern for one observation period, we propose a method for extending t-patterns to multiple observation periods. We attempt to algorithmically incorporate information from distinct observation periods and find t-patterns in the aggregate data while keeping in mind that they may be generated by heterogeneous processes.

A naïve approach might be to concatenate observation periods and perform calculations on all instances of candidate patterns in all observation periods, as long as they don’t stretch over the boundary of one observation period into another. However, this approach obscures important information that is provided by having separate observation periods. Subtly, this makes an extremely unfortunate simplifying assumption in estimating the probability of success for a single Bernoulli trial in the binomial distribution that ultimately determines the p-value for a potential t-pattern *TP*. There, we divide the count of occurrences of the right sub-pattern *TPR* by the total number of units of time to calculate the probability of *TPR* occurring at any one time according to our null hypothesis of uniformity. If we take this naïve approach, the probability of *TPR* occurring after an instance of the left sub-pattern *TPL* will be identical for all instances of *TPL* in the data.



To ameliorate some of the problems associated with the previous approach, we instead calculate the number of instances of *TPR* and for the observation period that the given instance of *TPL* belongs to. Thus, the null hypothesis becomes that the instances of *TPR* *within the observation period* are uniformly distributed *within the observation period*, as opposed to across all observation periods. Intuitively, if *TPR* is more common in observation period *i* than observation period *j*, the probability of seeing a *TPR* after a *TPL* should be higher in *i* than *j*, too.



**4.2 Changes to the Distribution**

This change to the model affects the distribution under the null hypothesis, and thus changes the calculation of the p-value for each *T* ∈ {candidate patterns}. In the original formulation,



where



In the revised formulation, might have instances in different observation periods. The distribution followed now becomes a Poisson binomial distribution where each Bernoulli trial is still independent but has a different probability of success. Specifically,



where is a vector whose entry is:



The subscriptdenotes that the count and total time refer to observation period *i* only. Unfortunately, the p-value calculation is complicated significantly by this change. If we observe instances, the p-value is defined as



But for each value in the summation, there are ways to get successes. Unlike in the binomial distribution’s calculation, since each trial has a different probability, we can’t use this as a multiplicative factor in an analytical solution. Instead, we must enumerate each possible way of getting successes in trials and calculate its probability separately. The p-value is then the sum of all of these probabilities for all values of . Heuristics for approximating the p-value without these costly operations acquire virtually identical results in practice.



**4.3 Heuristics for Calculating the p-value**

To avoid calculating these probabilities and summing them to get the p-value of one candidate pattern, we can approximate the Poisson binomial distribution with a binomial distribution with parameter. This approximation becomes less accurate as the variance of the probability vector increases, but drastically reduces the number of calculations necessary for each candidate pattern.



Another consideration is the case of an observation period with an instance of but no instances of . Our estimate of would be 0; there was no chance of observing after because there was no chance of observing anywhere. This could be a valid assumption if observation periods are segregated in such a way that only certain segments of the population have some event types, while others have other event types. In the case where each observation period represents a single sample from a roughly unchanging distribution, not occurring in any individual one is attributable to sampling error. With this in mind, a pseudocount can be added to the probability estimation:



**5. Observation Periods with Gaps**

The original formulation of the null hypothesis stated that the occurrences of are uniformly distributed across the observation period. In many real-world applications, this assumption is violated so badly that the results are meaningless. For example, consider the following observation period:



All of the events are bunched into two particular areas with a large gap in between. Assuming uniformity misses that there are some time units with zero probability of producing any event type. The time units in the periods of activity would then be underestimated by whatever probability mass is attributed to the gapped times.

Our method of more accurately modeling the underlying process is to model the period of activity as not across the entire observation period, but as being within some window *w* of an actual event instance. For an observation period, instances of *occurring within w time units of an instance of* are assumed to be uniformly distributed over *the intervals within w time units of an instance of* . Using the example observation period above:



The null hypothesis would estimate that the two instances of are uniformly distributed over the time units in the two windows instead of the time units in the whole observation period. This changes only the calculation of each to be



where is the modified count function which only considers events within the window of an instance of .



In practice, this parameter can be tuned to the specific application. For an application where there are days that are highly active but possibly separated by months of little or no activity at all, this extension allows t-patterns to be discovered at the hour level of detail for the active days without being obfuscated by the months of inactivity.

**6. Algorithms and Data Structures**

We first present a method of implementing the t-pattern discovery algorithm here. It is an exact method, preferred for cases where the number of event types and the number of observation periods is relatively small. We later discuss heuristics for the possibly more likely scenario where either of these numbers is very large. Each assumes the event types, , are known.



The algorithm is summarized by the following pseudocode:

CANDIDATE PATTERNS <- LIST OF CANDIDATE PATTERNS

FOR EACH EVENT TYPE :



FOR EACH EVENT TYPE :



ADD EVENT TYPE TO LIST OF CANDIDATE PATTERNS



WHILE THERE IS A CANDIDATE PATTERN TO CHECK:



CREATE DISTRIBUTION OF INTERVALS BETWEEN ,



CALCULATE P-VALUE FOR DISTRIBUTION

IF SIGNIFICANT:

ADD TO LIST OF EVENT TYPES



ADD ALL NEW POSSIBLE CANDIDATE PATTERN DISTRIBUTIONS WITH



AS OR



COMPLETENESS COMPETITION FOR PATTERNS FOUND

Initially, a candidate pattern is generated for each possible pair of event types, so there are of them. Then, for each of these patterns of length two that is determined to be significant, it is added to the set of event types and two candidate patterns of length three are added for each event type in the set (one where it is, one where it is ).



The search space here can be immense, especially if many significant patterns exist in the data, rapidly increasing and thus increasing the number of candidate patterns generated for each additional significant pattern found. Luckily, the search space is both sparse and contains many statistically significant patterns that will not be informative for the classification task. We exploit these attributes later by forming heuristics guiding our search away from the candidate patterns that offer little hope of being useful.



**7. Heuristics for Searching**

**7.1 Prune Infrequent Event Types**

It is easy to see how quickly the time and space requirements grow in the exact method. One method of increasing performance is by using heuristics to guide search through the state space, as opposed to using exhaustive search. One heuristic we propose is motivated by a method for mining frequent itemsets, the Apriori algorithm.

In the Apriori algorithm, we first define a parameter *k*, the support threshold. We then define as *frequent* only those itemsets of size 1 (single items) with at least *k* instances. We proceed by considering *frequent* only those itemsets of an arbitrary size that both have at least *k* instances and are made up exclusively of *frequent* itemsets of smaller sizes.

We can utilize this bottom-up approach in one of two ways. In the first, we trim infrequent event types that don’t meet the support threshold, and then further only add a candidate pattern to the list of event types if both its p-value is significant and its number of occurrences exceeds the support threshold. An alternative approach is to only consider as frequent the top *k* candidate patterns of each size.

**7.2 Prune Infrequent Event Type Pairs**

Similarly to the previous method, we can use a support threshold to avoid considering candidate patterns that occur infrequently. Instead of trimming individual event types, though, we trim pairings of event types that do not co-occur in the same observation period often. If our event types are partitioned in such a way that those in one group tend to only occur in observation periods of one type while those in another group occur in those of another type, we need not consider a candidate pattern with in the first group and in the other.



**7.3 Prune Uninformative Event Types**

Another approach to limit the search space is to only consider event types that appear locally useful for the ultimate goal, classification. We can do this by comparing the distribution of the class variable for those observation periods where this event type occurs against a null, uninformed model to calculate the information gain associated with the presence of this t-pattern. In this way, we can avoid unnecessarily exploring both areas of the search space with very high-order t-patterns and the many small, spurious patterns that inevitably arise from large data sets.

First, we calculate the empirical distribution of the class variable for the observation periods where the event type under consideration occurs. We then compare this to a multinomial distribution using the empirical distribution of the class variable for all observation periods. To perform this comparison, we use as a metric the Kullback-Leibler divergence (KL-divergence) between the two distributions. Here, the KL-divergence measures the information lost when approximating one distribution (filtered by the event type occurring) with another (the null, unfiltered full set of observation periods).

It has been shown that the KL-divergence, under certain assumptions, follows as a chi-squared distribution with one degree of freedom. Thus, we pick an appropriate value from this distribution to allow a reasonable number of event types through while still filtering for the presence of some information gain.

**8. Classification**

Finally, we consider the task of predicting the value of a class variable for each of the observation periods. For each t-pattern discovered, we create an indicator variable for the presence of it in each observation period. We can then feed these indicators as input into the classification approach of our choice. Factors affecting which approach to use include:

* Many t-patterns consist of the same or related sub-patterns, so these variables are highly correlated
* If we don’t filter on KL-divergence, many of these t-patterns are likely to be meaningless for the task

With these considerations in mind, models that are heavily parameterized or impose strong prior knowledge on the data might not be preferred.

**9. Conclusion**

In this paper, we have explained the ideas behind an original model for finding temporally related patterns in a single observation period. We then proposed a method for extending this to multiple observation periods, gathering information across independently generated units. We also discussed solutions to problems arising when the assumptions of the original model are heavily violated. The large computational load of pattern mining creates the need for guided searching, so we introduced possible heuristics to use to narrow the search space. Lastly, we incorporated the information from the presence, within one observation period, of the t-patterns we found across all observation periods into traditional classification methods.