Generative Modeling of Itemset Sequences derived from Real Databases

Rui Henriques[1] and Claudia Antunes[2]
[1]*KDBIO, Inesc-ID, Instituto Superior Técnico, University of Lisbon*
[2]*Dep. Comp. Science and Eng., Instituto Superior Técnico, University of Lisbon*
{*rmch,claudia.antunes*}*@tecnico.ulisboa.pt*

Keywords: Hidden Markov Models, Itemset Sequences, Real-World Databases

Abstract: The problem of discovering temporal and attribute dependencies from multi-sets of events derived from real-world databases can be mapped as a sequential pattern mining task. Although generative approaches can offer a critical compact and probabilistic view of sequential patterns, existing contributions are only prepared to deal with sequences with a fixed multivariate order. Thus, this work targets the task of modeling itemset sequences under a Markov assumption. Experimental results hold evidence for the ability to model sequential patterns with acceptable completeness and precision levels, and with superior efficiency for dense or large datasets. We show that the proposed learning setting allows: *i)* compact representations; *ii)* the probabilistic decoding of patterns; and *iii)* the inclusion of user-driven constraints through simple parameterizations.

1 Introduction

Recent work in the area of data mining reveals the importance of defining learning methods to simultaneously mine temporal and cross-attribute dependencies in real-world data (Henriques and Antunes, 2014). For this purpose, multi-dimensional and relational structures have been mapped as itemset sequences, temporally ordered sets of itemsets. This turns the mining of itemset sequences applicable not only to transactional databases like market basket analysis, but also over multi-dimensional databases as observed in healthcare and business domains. However, the common option to explore itemset sequences, sequential pattern mining (SPM), has not been largely adopted due to its voluminous results and parameterization needs. Additionally, although generative approaches allow for effective pattern-centered analyzes of multivariate sequences of fixed order, there are not yet experiments that show whether or not they can be extended to consider itemsets of varying length (cross-attribute occurrences) with acceptable performance. In this work we rely on parameterized hidden Markov models (HMMs) to deliver a compact and generative representation of sequential patterns that combine frequent co-occurrences (intra-transactional analysis) and precedences (inter-transactional analysis).

With the goal of overcoming the critical problems of traditional SPM methods, some approaches rely on compressed representations or define a deterministic generator of sequential patterns (Mannila and Meek, 2000). However, they can still grow exponentially. Additionally, these methods cannot disclose the likelihood of a pattern to be generated when assuming underlying noise distributions. To tackle these drawbacks, formal languages and HMMs have been applied to solve SPM task (Chudova and Smyth, 2002; Laxman et al., 2005; Jacquemont et al., 2009). However, these generative solutions are not able to model itemset sequences and depend on restrictive assumptions regarding the size, shape and noise of patterns.

This paper answers the question: to which extent can HMMs address these challenges. To answer we, first, propose solutions based on alternative Markov-based architectures. Second, we evaluate their performance by assessing the efficiency and the output matching against deterministic outputs for synthetic data and real databases. To the best of our knowledge, this is the first systematized work on how to use HMMs over multivariate symbolic sequences.

This paper is structured as follows. In *Section 2*, generative SPM is formalized and motivated. *Section 3* describes the proposed solutions. Results are provided in *Section 4* and their implications synthesized.

2 Background

Recent research shows that real-world databases can be expressively mined by mapping them as sets of itemset sequences (Henriques et al., 2013). Sequential pattern mining (SPM), originally proposed by (Agrawal and Srikant, 1995), still is a default option to explore itemset sequences.

Let an item be an element from an ordered set Σ. An *itemset I* is a set of non-repeated items. A *se-*

quence s is an ordered set of itemsets. A sequence a=$a_1...a_n$ is a *subsequence* of b=$b_1...b_m$ ($a \subseteq b$), if $\exists_{1 \leq i_1 < .. < i_n \leq m}$: $a_1 \subseteq b_{i_1}$,..,$a_n \subseteq b_{i_n}$. A sequence is *maximal*, with respect to a set of sequences, if it is not contained in any other sequence of the set. The illustrative sequence s_1=$\{a\}\{be\}$=$a(be)$ is contained in s_2=$(ad)c(bce)$ and is maximal w.r.t. S=$\{ae,(ab)e\}$.

Definition 1. *Given a set of sequences S and some user-specified minimum support threshold* θ*, a sequence* $s \in S$ *is* frequent *if contained in at least* θ *sequences. The* ***sequential pattern mining*** *task aims for discovering the set of maximal frequent sequences (*sequential patterns*) in S.*

Considering a database S=$\{(bc)a(abc)d, a(ac)c, cad(acd)\}$ and a support threshold θ=3, the set of maximal sequential patterns for S under θ is $\{a(ac), cc\}$. Traditional SPM approaches rely on prefixes and suffixes, subsequences with specific meanings, and on the (anti-)monotonicity property to deliver complete and deterministic outputs. However, these outputs are commonly highly voluminous and the frequency is a deterministic function (cannot flexibly consider underlying noise distributions).

Alternatives have been proposed, with a first class focused on formal languages and on the construction of acyclic graphs that define partial orders and constraints between items (Guralnik et al., 1998; Laxman et al., 2005). Probabilistic generative models as neural networks, hidden Markov models (HMMs) and stochastic grammars hold the promise to deliver compact representations given by the underlying lattices (Ge and Smyth, 2000). The expressive power, simplicity and propensity towards sequential data of HMMs turn them an attractive candidate.

Definition 2. *Consider a discrete alphabet* Σ*, a* ***first-order discrete HMM*** *is a pair* (T,E) *that defines a stochastic finite automaton where a set of connected hidden states* X=$\{x_1,..x_k\}$ *is expressed by a probability transition matrix* T=(t_{ij})*, with observable emissions described by probability emission matrix* E=$(e_i(\sigma)) = (e_{i\sigma})$*, where 1<i<k, 1<j<k and* $\sigma \in \Sigma$*.*

Under a first-order Markov assumption, emissions depend on the current state only. Let the system be in state x_i: it has a probability t_{ij}=$P(x_j|x_i)$ of moving to x_j state and probability $e_{i\sigma}$=$P(\sigma|x_i)$ of emitting σ item. (T,E) defines the HMM *architecture*.

Preferred emissions and transitions (paths with higher generation probability) are usually associated with regions that may have structural and functional significance. For specific architectures, different patterns such as periodicities or gap-based patterns can be revealed by analyzing the learned (T,E) parameters (Baldi and Brunak, 2001). Based on this observation, alternative Markov-based approaches have been proposed for the mining of patterns using different: *i)* task formulations, *ii)* assumptions, and *iii)* learning settings (Chudova and Smyth, 2002; Ge and Smyth, 2000; Laxman et al., 2005; Murphy, 2002).

The commonly target *tasks* include the discovery of generative strings[1] as consensus patterns and profiles (across a set of sequences) or motifs (within one sequence). These tasks have been mainly applied to univariate sequences (Chudova and Smyth, 2002; Ge and Smyth, 2000; Fujiwara et al., 1994; Murphy, 2002), with some exceptions allowing numeric sequences with a fixed multivariate order (Bishop, 2006) and graph structures (Xiang et al., 2010). Additionally, the majority is centered on the discovery of contiguous items, not accounting for items' precedences of arbitrary distance.

Previous work by (Laxman et al., 2005; Jacquemont et al., 2009; Cao et al., 2010) provide important principles for the decoding of sequential patterns but both fail to model co-occurrences.

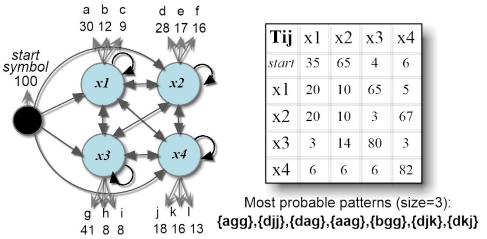
What makes the problem difficult is that few is known a priori about what these patterns may look like. Typically, the number and disposal of precedences and co-occurrences can significantly vary across patterns. State-of-the-art approaches (Chudova and Smyth, 2002; Murphy, 2002) place assumptions regarding the type, length and number of patterns, and commonly assume that patterns do not overlap. These restricted formulations require background knowledge that may not be available.Even so, traditional learning settings of HMMs may still present significant additional challenges to pattern-based tasks. One of them is the convergence of emission probabilities. The spurious background matches in long sequences can lead to false detections, making pattern discovery difficult. The Viterbi algorithm alleviates this problem (Bishop, 2006) but does not guarantee the convergence of emission probabilities. In literature, three *learning settings* have been proposed. (Murphy, 2002) requires emission distributions to be (nearly) deterministic, i.e., each state should only emit a single symbol, although this symbol is not specified. This is achieved using the minimum entropy prior (Brand, 1999)[2]. An alternative is to use a mixture of Dirichlets (Brown et al., 1993). Finally, Chudova (Chudova and Smyth, 2002) introduces a Bayes error framework.

3 Technical Solutions

In this section we propose solutions to mine sequential patterns using HMMs without the need to

[1] Given an alphabet Σ, a *generative string* is a distribution over Σ allowing substitutions with noise probability ε

[2] Assuming e_j to be multinomial emissions for a x_j state, entropy is given by: $H(e_j)$=$-\Sigma_\sigma e_{j\sigma} log(e_{j\sigma})$



Tij	x1	x2	x3	x4
start	35	65	4	6
x1	20	10	65	5
x2	20	10	3	67
x3	3	14	80	3
x4	6	6	6	82

Most probable patterns (size=3):
{agg},{djj},{dag},{aag},{bgg},{djk},{dkj}

Figure 1: Pattern mining with fully inter-connected arch.

rely on assumptions. We overview existing architectures, propose a simple data mapping for their application over itemset sequences, define initialization, decoding and learning principles, and, finally, incrementally propose more expressive architectures.

3.1 Existing HMM Architectures

The paradigmatic case of pattern mining over univariate sequences is to use fully inter-connected architectures (Baldi and Brunak, 2001), illustrated in Fig.1, plus an efficient method (based on propagation on graphs or dynamic programming) to exploit the model. Such method can retrieve patterns of arbitrary length by exploiting the most probable transitions and emissions. The criteria of whether is frequent or not may either depend on the ranking position of the pattern or on the overall generation probability.

Although the fully inter-connected architecture can be always used as a default option, in order to minimize the introduced problem of emissions' convergence there are alternative HMM architectures with sparser connectivity. A basic architecture adopted for motif discovery in univariate sequences is one that explicitly models the pattern shape and uses a distribution that either guarantees the convergence of emission probabilities, such as in (Murphy, 2002), or allows for a fixed error threshold, such as in (Chudova and Smyth, 2002). An important assumption for these architectures, illustrated in Fig.2, is whether the target pattern emerge from the supporting sequences or/and from the recurrence of the pattern within each sequence. For this case, a new state and transitions $\{t_{45}, t_{55}\}$ need to be included, and $x_4 \rightarrow x_1$ transition deleted (t_{41}=0). Additionally, transition probabilities, T, need to be carefully initialized. For instance, the self-loop transitions encode the expected length of the inter-pattern segments by using, for instance, a geometric distribution.

However, this architecture does not support patterns with non-contiguous items. For the discovery of more relaxed patterns, Left-to-Right Architectures (**LRA**s) (Baldi and Brunak, 2001; Liu et al., 1995) are commonly adopted in biology and speech recognition. LRAs consider both insertions (to allow sparsity) between pattern items and deletions (skip states) characterized by void emissions. Deletions can be used both to skip noisy occurrences or discover patterns with reduced length. With LRAs, a large number of precedences can be retrieved through the analysis of emissions along the main path. These emissions can be though as the set of symbols for aligning sequences. Fig.3 illustrates this architecture. Uniform initialization of transition probabilities without a prior that favors transitions toward the main states should be avoided in order to guarantee that main states are selected (instead of only insert-delete combinations).

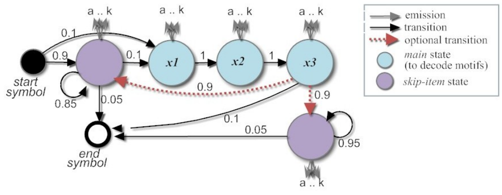


Figure 2: Motif discovery with shape-specific architecture

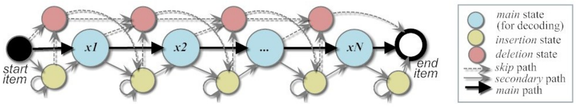


Figure 3: LRA: precedences of arbitrary length

3.2 Proposed Solutions

To be able to process itemset sequences under a Markov assumption, we propose a simple mapping of each sequence of itemsets introducing a special symbol for delimiters, $\Sigma \cup \{\$\}$. Illustrating, a sequence of itemsets $(ab)ca(ac)$ is now mapped into a univariate sequence $\$ab\$c\$a\$ac\$$, where \$ is the symbol that delimits co-occurrences.

Under this mapping, we can apply existing HMM architectures prepared to deal with univariate sequences. The retrieval of patterns from the underlying lattices results in combined sequences of both regular items and delimiters. Empty itemsets (sequent delimiters) are removed from the decoded patterns.

3.2.1 Structural principles

Initialization principles. To define the initialization of *transition probabilities*, **T**, we propose the use of metrics based on simple statistics over the dataset S and on pattern expectations. Fig.2 shows the initializations for motif-oriented architecture where the average length of sequences is 20 items, and where the probability of visiting an item belonging to a pattern is α=33,(3)% based on the expected average pattern size (3 items) and recurrence (9 inter-pattern items). When alternative paths are available (as in fully-interconnected architectures), a random weight

(ε>0) can be added to each transition in order to facilitate the learning convergence.

Finally, the *emission probabilities*, **E**, should be equal for all the items in order to not bias the learning. The emission probabilities of delimiters should: *i)* slightly increase for fully-interconnected architectures in order to guarantee the distinction between precedences and co-occurrences, and *ii)* slightly decreased for LRAs to avoid that all of the main path emissions converge to the delimiter symbol.

Learning settings. The distribution underlying the learning of emissions probabilities must guarantee strong convergence of emissions for the accurate decoding of patterns, but simultaneously avoid a too strong convergence that limits the coverage of interesting patterns. The use of entropy or Dirichlet distributions offer too strong convergence that restricts the decoding of multiple patterns along the model paths, while the use of the Bayes error rate (Chudova and Smyth, 2002) was observed to be difficult for patterns with significant autocorrelation and with a periodic structure (as *bcbcbc* or *aaaa*). The use of Viterbi (Bishop, 2006) is a good compromise since it guarantees the learning convergence without requiring a too restrictive number of eligible emissions per state.

Decoding principles. Beyond the definition, initialization and learning of the generative model, there is the need to define principles for a robust and efficient decoding of patterns from lattices. First, the use of the anti-monotonic property to prune paths. Second, the use of probability thresholds that traduces the criterion that defines whether a pattern is or not frequent. This threshold should be weighted by the length of the pattern to avoid a bias towards small patterns.

3.2.2 Extending the existing architectures

Although the existing architectures can be applied as-is using the proposed data mapping and parameter initializations, they are prone to decoding errors. Note the case where a state emits a reduced number of items with high probability, but one of these items is the itemset delimiter. For this common case, the distinction between precedences and co-occurrences becomes blurry, in particular if the state has a self-loop transition (as in fully-interconnected architectures). To improve accuracy, we propose new architectures with dedicated states to emit delimiters.

Before introducing them, consider an extension of LRAs where main and insert states only emit regular items, delimited by two states that can only emit the delimiter item with transitions to self-looping states. This architecture, referred as **CoIA** (Co-occuring Items Architecture) and illustrated in Fig.4, captures intra-transactional patterns by seeing each itemset as a univariate sequence. A transition to the initial state can be used to consider the pattern recurrence within a sequence.

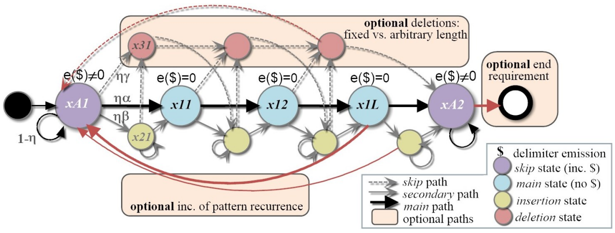


Figure 4: CoIA: discovery of co-occuring items

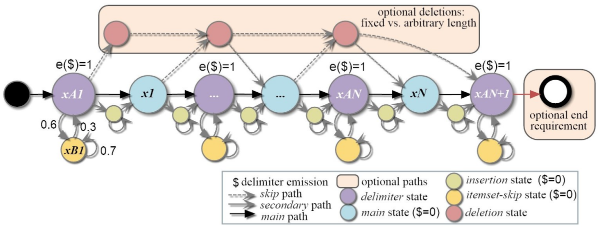


Figure 5: IPA: discovery of inter-transactional patterns

Two variations can be considered over this architecture. First, an end state can be linked to the architecture. This guarantees that, at least, one intra-transactional pattern per sequence is used to learn the left-to-right emissions. The end state can be implemented by adding an ending symbol at the end of each input sequence. Second, deletion states can be removed. This turns the intra-transactional patterns of fixed length, which simplifies the learning, although it restricts the original potential for decoding patterns of arbitrary length along the main path.

Now consider the proposed Itemset-Precedences Architecture (**IPA**) illustrated in Fig.5. With this architecture we can model inter-transactional patterns by decoding learned emissions along the main path. Two aspects of IPA should be noticed. First, insert states are used to remove non-frequent items. Second, each state dedicated to emit delimiters has a transition to a self-looping state in order to allow for gaps between itemsets. In this way, we transit from contiguous items to item precedences.

Beyond not capturing intra-transactional patterns, IPA suffers from another drawback. Since there is no guarantee that most of the input sequences will reach state x_p, the significance of precedences decoded from the first portions of the path is greater than from the last portions. This turns the pattern decoding algorithms more complex as they need to reduce the tolerance (cut-off thresholds) along the path in order to avoid the output of patterns prone to errors.

Similarly to the CoIA architecture, we can consider a variation of IPA that includes deletion states,

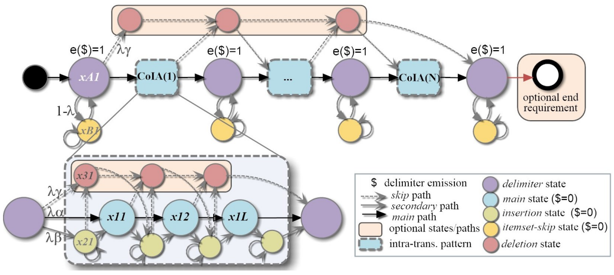


Figure 6: SPA: discovery of sequential patterns

which allows for a well-distributed level of significance for the learned probabilities across regions.

Finally, for the learning of generative models that combine precedences and co-occurrences, we propose the integration of the previous CoIA and IPA architectures. A CoIA architecture is applied between each sequent pair of delimiter states from the IPA architecture. A simplification of the resulting architecture, sequential patterns architecture (SPA), is illustrated in Fig.6. SPA is prone to deliver shapes as the one described by the $(ab)(bcd)a$ pattern.

The number of discovered precedences can be arbitrary by introducing deletion states across delimiter states. The size of the intra-transactional patterns can be also arbitrary (unless the user is interested in specific pattern shapes) by considering deletion states within each CoIA component. Finally, the end requirement or the recurrence within each sequence (loop to initial state) are possible variations.

Beyond reducing the probability of decoding patterns that are not frequent, SPA has a very efficient decoding step. There is only the need to analyze combinations of emissions along the main path.

3.2.3 Convergence of emissions

A critical drawback of previous SPA, IPA and CoIA architectures is that they cannot model a large number of patterns. These architectures rely on one main path only, where each state commonly emits a reduced number of items with significant probability, which commonly results in a compact set of patterns. This turns this method to be not competitive with deterministic peers and, therefore, of limited utility. Although one can reduce the threshold probabilities of the decoding phase or decrease the convergence threshold of the adopted HMM learning algorithm (to relax the convergence of emissions), this significantly degrades the quality of the output patterns.

A simple solution to avoid this problem is to adopt an iterative scheme, where each iteration is composed of three phases – learning, decoding, and masking of patterns – until patterns cannot be further decoded.

We propose an alternative solution that relies on multiple paths, so the sum of the compact sets of patterns from each path approximates the true number of frequent patterns. The number of paths can be defined by dividing the expected number of frequent patterns by the average number of patterns able to decoded from each SPA component.

4 Results

The target hidden Markov models[3] were adapted from the HMM-WEKA extension prepared for classification (implemented according to (Bishop, 2006; Murphy, 2002) sources). The extensions were implemented using Java (JVM version 1.6.0-24) and the following experiments were computed using an Intel Core i5 2.80GHz with 6GB of RAM.

We adopted synthesized datasets based on IBM Generator tool[4] by fixing values for all, except one, of the parameters, and by varying the value for the remaining parameter. The default dataset contains m=2.500 sequences, with an average of n=10 transactions each, each transaction with l=4 items on average. The alphabet has 1.000 items. The average length of maximal patterns is set to 4 and maximal frequent transactions set to 2. The values for different sequential patterns and transactional patterns were set to 1.000 and 2.000, respectively. This default setting generates near 10.000 sequential patterns for a support of 1% (with the majority of them having more than 5 items), and more than 400 sequential pattens for a support of 4%. The varied parameters include the number of items per itemsets, the number of itemsets per sequence, and the number of available items (density). These combinatorial set of datasets were tested for the architectures introduced in previous section, whose properties are illustrated in Table 1.

In order to validate if the proposed solutions have an acceptable performance, it is critical to assess *efficiency* of the learning-and-decoding stages against

Architecture	*Properties*
Fully Interc.	10 states. Delimiter emissions allowed on every state.
LRA	Main path with 14 states (max number of items and delimiters in pattern using expectations). Delimiter emissions allowed on main and insertion states.
SPA (no end requirement)	Max N precedences with max L co-occurrences each (N=8 and L=4 are the default data expectations).
SPA (end requirement)	Max N precedences with max L co-occurrences each (N=8 and L=4 are the default data expectations).
Multi-path SPA	Three SPA (end requirement) paths.

Table 1: Tested HMM architectures

[3] Software: http://web.ist.utl.pt/rmch/software/hmmevoc
[4] http://www.cs.loyola.edu/∼cgiannel/assoc_gen.html

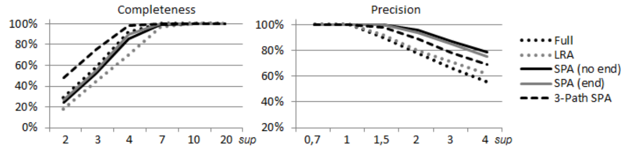


Figure 7: Completeness and precision of alternative architectures for varying minimum support

traditional SPM approaches and their ability to extract all frequent patterns (completeness) and only those ones (precision) (Jacquemont et al., 2009). These metrics provide an understanding on whether is it possible to decode patterns from compact generative models that match the output of deterministic approaches. *Completeness* is the fraction of frequent sequential patterns that were decoded by our approach. *Correctness* or *precision* is the fraction of decoded sequential patterns that are also retrieved by deterministic miners.

$$Completeness = \frac{|GenerativeOutput \cap DeterministicOutput|}{|DeterministicOutput|}$$

$$Precision = \frac{|GenerativeOutput \cap DeterministicOutput|}{|GenerativeOutput|}$$

The provided results for these metrics are an average using 10 datasets per parametrization. Additionally, we statistically tested the significance of the observed differences by using a *paired two-sample two-tailed t-Student test* with 9 degrees of freedom.

4.1 Completeness-Precision

An initial view on how the generative outputs compare against deterministic outputs for varying levels of support is depicted in Fig.7. When assessing these results against the number of frequent patterns of the dataset, two major observations can be derived. First, our generative approach is able to cover all frequent itemsets with support above ∼5%, and the completeness level degrades for lower supports (2% support delivers >3000 frequent patterns) although the larger architectures can still cover a large number of frequent patterns (>1500). Second, precision levels are 100% above 1% of support, which means that our approach is able to only deliver patterns whose frequency is >1% – important since generative approaches allow for noisy occurrences. Multi-path SPA is significantly superior than remaining options in terms of completeness, while all the proposed architectures were significantly better than traditional fully-interconnected and LRA architectures in terms of precision.

Completeness. Fig.8 illustrates the completeness of the proposed architectures to capture patterns with support above 4%. Note that an increase of support to 6% results in an approximated levels of 100% for all the architectures across datasets.

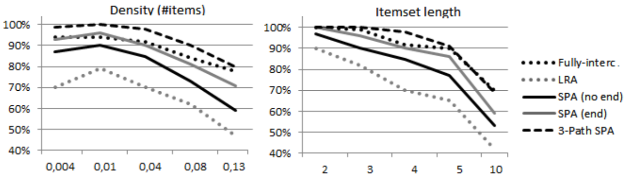


Figure 8: Completeness of alternative architectures against parameterizable datasets (θ=4%)

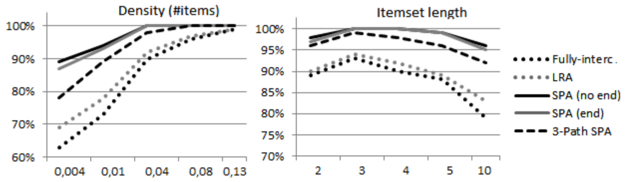


Figure 9: Precision of alternative architectures against parameterizable datasets (θ=1.5%)

Two major observations result from the analysis. First, multi-path and fully-interconnected architectures achieve a good completeness since they can focus on different subsets of probable emissions along the alternative architectural paths. Second, the levels of completeness degrade for higher densities and itemset length. This is a natural result of the explosion of patterns discovered by deterministic approaches under such hard settings. Note that an increase of multi-path SPA to six paths is able to hold completeness levels above 96% for all the adopted settings.

Precision. Fig.9 illustrates the precision of the proposed architectures, that is, the fraction of decoded patterns deterministically frequent (support higher than 1.5%). Note that a decrease of support to 0.8% results in an approximated levels of 100% for all the architectures across datasets. Generative approaches hold high levels of precision (>90%) across the majority of dataset settings. There is a slight decrease of precision for short itemsets since the number of deterministic patterns is smaller than the average number of decoded patterns and for large itemsets due to a cumulative decoding error associated with larger patterns and the intra-transactional size constraints adopted for SPA architectures. Additionally, the observed decrease in precision for high levels of sparsity is not only explained by a reduced set of deterministic patterns (potentially smaller than the decoded set) but also by an intrinsic difficulty to guarantee the convergence towards a reduced set of emissions that can block larger decoded outputs. Finally, fully-interconnected and LRA architectures are not as competitive as SPA-based architectures due to the additional error propagation associated from not constraining delimiter emissions to dedicated states.

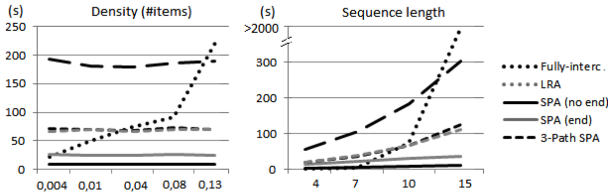


Figure 10: Efficiency against parameterizable datasets

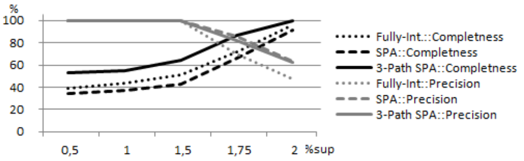


Figure 11: Completeness-precision for Foodmart dataset

4.2 Efficiency

The comparison of efficiency for the alternative architectures against PrefixSPAN[5] (Pei et al., 2001), one of the most efficient deterministic SPM algorithms, applied with a support threshold of 1% is illustrated in Fig.10. Under this threshold, the number of deterministic frequent patterns vary between 1.000 patterns (sparser and smaller datasets) to near 100.000 patterns (denser and larger datasets).

Generative approaches are particularly suitable over dense datasets against deterministic approaches, whose performance rapidly deteriorates for densities above 10%. Datasets with densities beyond 20% are very common across a large number of domains. Interestingly, the performance of the generative approaches does not significantly change with varying densities. This is explained by a double-effect: learning convergence deteriorates with an increased density, but this additional complexity is compensated by a higher efficiency per iteration since there is a significantly lower number of emission probabilities to learn per state.

Additionally, generative approaches scale better with an increased number of itemsets per sequence than PrefixSPAN. Under the default settings, PrefixSPAN is only efficient for sequences with less than 15 itemsets. Understandably, fully-interconnected and LRA are the most efficient solutions due their inherent structural simplicity.

4.3 Real-world databases

Our approach was applied in sparse and dense real-world databases. Fig.11 illustrates the performance of our approach over itemset sequences derived from the Foodmart data-warehouse[6]. Each itemset sequences is composed of temporally ordered basket sales from a specific customer between 1997-98 (average of 6 items per basket and 6 baskets). Two important observations result from this analysis. First, the best Markov-based architectures are able to achieve 100% precision levels while still being able to recover more than half of the frequent patterns for very low levels of support (θ=0.5%). Second, for medium levels of support (2%), our generative approach is able to cover all the frequent patterns, although it additionally delivers patterns with a lower support (1.5-2%) that can penalize the precision.

Secondly, we applied our approach over the dense *Plan* dataset[7], which is not tractable for deterministic SPM approaches even when considering a constrained number of instances (<1000). In line with previous efficiency observations, our generative alternatives were able to learn emission and transition probabilities in useful time. In fact, our approach is critical for similar dense cases as it is able to decode patterns based on the most accentuated probability differences across the learned lattices.

4.4 Discussion

Generative SPM approaches provide more scalable principles than deterministic peers to deal with dense datasets and with very large sequences. Even when considering complex architectures, generative approaches tend to perform better in terms of efficiency for dataset with these properties. Additionally, the analyzed precision-completeness levels is over 90% for the most expressive architectures across settings, which guarantees the deterministic significance of the decoded patterns. These are particularly attractive levels knowing that the probabilistic learning of patterns accounts for noisy occurrences that can lead to a substantially different (but similarly interesting) output. The observed performance shows that compact representations of large outputs is possible for SPM over itemset sequences using generative models.

Additionally, these models present three intrinsic properties of interest. *First*, they provide a probabilistic view for the sequential patterns' support that accounts for occurrences under noise distributions. *Second*, the probability of generating any sequential pattern can be assessed on a query-basis. Note that deterministic approaches only disclose support for the frequent patterns. *Third*, the introduction of background knowledge and constraints, as the selection of specific pattern shapes in accordance with domain expectations, can be easily incorporated in the mining process by parameterizing the target architecture.

[5] http://www.philippe-fournier-viger.com/spmf/

[6] https://sites.google.com/a/dlpage.phi-integration.com/pentaho/mondrian/mysql-foodmart-database

[7] http://www.cs.rpi.edu/∼zaki/software/plandata.gz

5 Conclusion

This article proposes a methodology for the definition of generative models under a Markov assumption to explore itemset sequences, an increasingly adopted data format to capture temporal and cross-attribute dependencies in real-world data. This allows a compact and probabilistic view of sequential patterns that tackles the problems of existing generative approaches, which can only deal with constrained formulations of the task. The methodology covers multiple architectures and learning settings that guarantee the relevance of the decoded patterns.

We show that the efficiency of the proposed SPM generative approaches is competitive with traditional SPM deterministic approaches on synthetic and real data. Additionally, the proposed approaches hold good levels of output-matching across a wide variety of synthetic datasets. This is considerably attractive since generative approaches offer a probabilist view of patterns where the notion of pattern relevance is rather different than the traditional counting support as it allows for noise distributions underlying pattern occurrences. This opens a new door for the generative formulation of SPM. This formulation holds the potentiality to deliver: compact representations of commonly large outputs; a probabilistic view of patterns (allowing for noise distributions, an alternative view of the traditional support); pruned searches under user-driven constraints; and a basis for query-driven decoding of patterns of interest.

Relevant future directions include: the assessment of changes in classifiers performance when adopting pattern-sensitive generative models; the study of the potential to dynamically self-learn expressive architectures from data; and the analysis of the impact of these generative models for a broader-range of real-world databases.

Acknowledgments

This work was supported by *Fundação para a Ciência e Tecnologia* under the project D2PM, PTDC/EIA-EIA/110074/2009, and the PhD grant SFRH/BD/75924/2011.

REFERENCES

Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *ICDE*, pages 3–14. IEEE CS.

Baldi, P. and Brunak, S. (2001). *Bioinformatics: The Machine Learning Approach*. Adaptive Comp. and Mach. Learning. MIT Press, 2nd edition.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Info. Science and Stat. Springer.

Brand, M. (1999). Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Comput.*, 11(5):1155–1182.

Brown, M., Hughey, R., Krogh, A., Mian, I. S., Sjölander, K., and Haussler, D. (1993). Using dirichlet mixture priors to derive hidden markov models for protein families. In *1st IC on Int. Sys. for Molecular Bio.*, pages 47–55. AAAI Press.

Cao, L., Ou, Y., Yu, P. S., and Wei, G. (2010). Detecting abnormal coupled sequences and sequence changes in group-based manipulative trading behaviors. In *ACM SIGKDD*, pages 85–94. ACM.

Chudova, D. and Smyth, P. (2002). Pattern discovery in sequences under a markov assumption. In *ACM SIGKDD*, pages 153–162. ACM.

Fujiwara, Y., Asogawa, M., and Konagaya, A. (1994). Stochastic motif extraction using hidden markov model. In *ISMB*, pages 121–129. AAAI.

Ge, X. and Smyth, P. (2000). Deformable markov model templates for time-series pattern matching. In *ACM SIGKDD*, pages 81–90. ACM.

Guralnik, V., Wijesekera, D., and Srivastava, J. (1998). Pattern directed mining of sequence data. In *ACM SIGKDD*, pages 51–57.

Henriques, R. and Antunes, C. (2014). Learning predictive models from integrated healthcare data: Capturing temporal and cross-attribute dependencies. In *HICSS*. IEEE.

Henriques, R., Pina, S. M., and Antunes, C. (2013). Temporal mining of integrated healthcare data: Methods, revealings and implications. In *SDM: 2nd IW on Data Mining for Medicine and Healthcare*. SIAM Pub.

Jacquemont, S., Jacquenet, F., and Sebban, M. (2009). Mining probabilistic automata: a statistical view of sequential pattern mining. *Mach. Learn.*, 75(1):91–127.

Laxman, S., Sastry, P., and Unnikrishnan, K. (2005). Discovering frequent episodes and learning hidden markov models: A formal connection. *IEEE TKDE*, 17:1505–1517.

Liu, J., Neuwald, A., and Lawrence, C. (1995). Bayesian models for multiple local sequence alignment and gibbs sampling strategies. *American Stat. Ass.*, 90(432):1156–1170.

Mannila, H. and Meek, C. (2000). Global partial orders from sequential data. In *ACM SIGKDD*, pages 161–168. ACM.

Murphy, K. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, CS.

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., and Hsu, M. (2001). Prefixspan: Mining sequential patterns by prefix-projected growth. In *ICDE*, pages 215–224. IEEE CS.

Xiang, R., Neville, J., and Rogati, M. (2010). Modeling relationship strength in online social networks. In *IC on World wide web*, WWW, pages 981–990. ACM.