Dense Neighborhood Pattern Sampling in Numerical Data

Arnaud Giacometti* Arnaud Soulet*

Abstract

Pattern mining in numerical data remains a challenging task due to the pattern search space that becomes potentially infinite with real-valued dimensions. Most approaches reluctantly reduced the expressiveness of mined patterns to make possible extraction. Despite this expressiveness loss, they do not provide results within a short response time of a few seconds. This paper addresses the instant discovery of patterns in numerical data based on sampling techniques. Instead of splitting each dimension into intervals, we use a metric to introduce the density as new interestingness measure, and to define neighborhood patterns. The language of neighborhood patterns is semantically rich but in return, its size is infinite. We then present a new exact and non-enumerative random procedure to sample this infinite language according to density. An experimental study demonstrates the good compromise between precision and diversity of neighborhood patterns. Finally, in the context of associative classification, we show that a sample of neighborhood patterns is as accurate as traditional methods that traverses the entire search space.

1 Introduction

During the last two decades, pattern mining has been a very active field of data mining by offering a large number of algorithms dedicated to more or less complex *qualitative* data such as itemsets, sequences or graphs [15]. However many application areas require the use of *quantitative* data such as spatial coordinates in geography, demographic data in economics and so on. In order to take into account numerical dimensions, the most popular approach is to partition them before applying the pattern mining algorithm (if there exists no natural partition). In general, the discretization [8] produces non-overlapping intervals meaning that each value of the same dimension is inside a single interval. It is well known that this technique induced an inherent loss of information on the distance between two values. For a given dimension, two very close values may be in separate intervals and be regarded as completely different. In contrast, two more distant points may be in the same interval and be regarded as identical. Moreover, these intervals without overlapping inevitably degrade the diversity of forthcoming mined patterns.

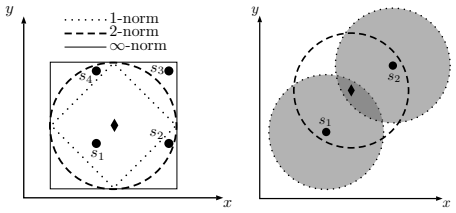

Figure 1: Impact of the norm on a neighborhood pattern in two dimensions (left) ; A diamond pattern in $\mathcal{L}(\mathbb{D})$ denser than the two patterns s_1 and s_2 in $\mathcal{L}(S)$ (right)

In order to overcome these limitations, this paper adresses the discovery of patterns in numerical data without considering any discretization process. To this end, we first introduce a language of patterns, called *neighborhood* patterns. Given a set of dimensions $\mathbb{D}$, containing either categorical or numerical values, a *neighborhood* pattern $x[D]$ is simply a point x in a subspace $D \subseteq \mathbb{D}$ for a p-norm and a radius r. Its set of neighbors is then defined as the subset of data points whose projection on D is at a distance lower than r. [20] has already used a dimensional point as pattern with a number of neighbors instead of a radius. As illustration, the left-hand side of Figure 1 presents 3 neighborhood patterns with a same center (represented by a diamond) for a same radius.

Due to the infinite domain for real-valued dimension, it is clear that the full-space language $\mathcal{L}(\mathbb{D})$ of neighborhood patterns is also *infinite*. Therefore, a complete enumeration method to extract the set of all interesting neighborhood patterns is not possible. One possible approach [20] is to limit the extraction to the dataset sub-language $\mathcal{L}(S) \subseteq \mathcal{L}(\mathbb{D})$ containing only neighborhood patterns $x[D]$ where x is the projection on D of a point in the dataset S. As this language is finite, it makes possible a complete enumeration but it misses the most relevant neighborhood patterns. As illustration, let us consider Figure 1 (right-hand side) representing a two-dimensional space with two data points s_1 and s_2. All points in the intersection of the neighborhoods of s_1 and that of s_2 (like the plotted diamond) has

*University of Tours, firstname.lastname@univ-tours.fr

two neighbors and satisfy a minimum threshold equal to 2. In contrast, each pattern in $\mathcal{L}(S)$ has exactly 1 neighbor because no projection of $s_{i\in\{1,2\}}$ has two data points in its neighborhood (i.e., no projection is contained in the darkest area). So there would be no patterns mined with 2 as minimum threshold. This example clearly illustrates the importance of addressing the full-space language $\mathcal{L}(\mathbb{D})$ instead of the dataset language $\mathcal{L}(S)$.

To deal with the infinite size of the full-space language $\mathcal{L}(\mathbb{D})$, this paper benefits from pattern sampling techniques introduced in [17, 4]. Basically, pattern sampling aims at drawing patterns in a language $\mathcal{L}$ with a probability proportional to an interestingness measure m. This approach has been successfully applied with different interestingness measures on discrete data. Its strength is to offer the user a fast and direct access to the entire pattern language and with no parameter (except possibly the sample size). This paper shows how to extend pattern sampling to neighborhood patterns considering the full-space pattern language and the density as interestingness measure. For instance, considering the example of Figure 1 (right-hand side), our method is twice as likely to draw a neighborhood pattern in the dark gray area as in the light gray area. It naturally focuses on *data vectors serving to describe an anomalously high local density of data points* as defined in [16].

The main contributions of the paper are as follows:

- We introduce a language of patterns, called *neighborhood* patterns, that allows the discovery of interesting pattern from numerical data without using any discretization process. We also formally define the density to evaluate the interestingness of neighborhood patterns.
- We propose an exact and non-enumerative sampling procedure addressing the infinite pattern language $\mathcal{L}(\mathbb{D})$. This method instantly returns neighborhood patterns according to a probability density function proportional to their density. We detail the sampling procedure for three different norms: 1-norm, 2-norm and ∞-norm.
- We present a large set of experimental results. On the one hand, we use two measures, plausibility and diversity, to assess the intrinsic quality of sampled patterns. On the other hand, we show how these patterns lead to build accurate associative classifiers.

The paper is organized as follows. Section 2 reviews some related work about pattern mining in numerical data and pattern sampling methods. Section 3 introduces basic definitions and the formal problem statement. Neighborhood pattern sampling algorithm is detailed in Section 4. We report a study on benchmarks in Section 5 evaluating the plausibility and the diversity of the approach, and the accuracy of classifiers based on neighborhood patterns. We conclude in Section 6.

2 Related Work

2.1 Pattern Structure for Numerical Data The introduction has already mentioned discretization methods [8]. There are some other transformation techniques of numerical data into binary data (e.g., in [1], the generated binary transaction encodes the neighborhood of the data point) that also lose information. To alleviate this problem, online partitioning approaches dynamically build intervals during the extraction of patterns [27, 14, 19] aiming at considering all possible intervals for each dimension. Unfortunately, this exhaustive approach is unfeasible in practice due to the prohibitive number of combinations much higher than the number of patterns in classical binary data. An elegant framework [19] significantly reduces the number of intervals by benefiting from condensed representation principles. However, this approach is insufficient to deal with large real-world datasets and the use of heuristics to not generate all the intervals remains necessary as done in [27, 14]. Again, the completeness paradigm is sacrificed to make feasible the mining task.

Finally, offline or online partitioning of numerical data are achieved dimension by dimension often ignoring the multivariate phenomena. There are few notable exceptions for two-dimensional spaces including [10]. Correlations concerning several dimensions will be then more difficult to identify [2]. Besides, as the combination of intervals (built individually on each dimension) form an hyperrectangle whose volume increases rapidly with the number of dimensions, the paradigm of interval is often more sensitive to outliers. In this paper, the notion of neighborhood with an infinite norm leads to the same topology (i.e., hyperrectangles). But it also makes possible to benefit from another topology like hyperspheres thanks to the 2-norm.

There are also other discrete pattern structures [7, 6, 18, 28] to circumvent the difficulties stemming from numerical data. These pattern structures remain sets of literals but their evaluation on the dataset benefits from the numerical nature of data. For instance, a gradual pattern [7] identifies a set of variations often observed between dimensions when comparing two data points (e.g., "the higher the age, the higher the salary") i.e., couples of a variation sense (ascending/descending) and a dimension. Other approaches [6, 18, 28] still mine sets of dimensions but they take into account the numerical data in the support calculation. An important difference is that for all of these proposals, a pattern is not

local, but global since the computation of its support always considers all the data points. Besides, all these pattern structures are discrete and in particular, they are not points in a data subspace. In contrast, a neighborhood pattern is truly local and numerical because it relies on a center involving numerical values.

2.2 Pattern Sampling Pattern sampling [17, 4] aims at accessing the pattern space $\mathcal{L}$ by an efficient sampling procedure simulating a distribution $\pi : \mathcal{L} \rightarrow [0, 1]$ that is defined with respect to some interestingness measure m, i.e., $\pi(.) = m(.)/Z$ where Z is a normalizing constant. As the pattern language is fully addressed proportionally to m, this approach guarantees a good variety of patterns returned to the user unlike heuristic approaches (including those whose goal is to find patterns maximizing interestingness criteria) and even, statistical properties [11]. As constraint-based pattern mining, pattern sampling problem has been proposed for different languages like itemsets [4] and graphs [17], and different interestingness measures including support [17, 4], area [4], discriminative measure [4], utility measure [4, 23]. Additional constraints are sometimes mandatory on the sampled patterns as it is the case in [9] that benefits from the SAT framework. But, to the best of our knowledge, no pattern sampling proposal addresses an infinite language and especially, large subspaces with numerical dimensions.

There are two main families of pattern sampling approaches. Markov Chain Monte Carlo (MCMC) method [17, 24, 3] uses a random walk on the partially ordered graph formed by the pattern language. With such a stochastic simulation, it is difficult to set the equilibrium distribution with the desired properties and the convergence to the stationary distribution within an acceptable error can be slow. In contrast, two-step random procedure [4, 23] samples patterns exactly and directly without simulating stochastic processes. Basically, this procedure randomly selects a data point according to a first distribution and then, it selects a pattern from this data point according to a second distribution. The choice of these two distributions enable a fine control of the produced patterns (e.g., area or discriminative measure as interestingness). This method is particularly effective for drawing patterns according to support or area (linear with the size of the dataset). But it turns out quadratic or worse for some measures (like the discriminative measure) requiring the drawing of several data points in the first step. In this paper, our proposal is based on the second family due to its efficiency. Nevertheless, we complete the two-step random procedure by a new and essential third step for taking into account the nature of numerical data.

3 Problem Statement

3.1 Preliminary Definitions We now introduce the formal framework of this paper. $\mathbb{D}$ is a set of dimensions. $dom(d)$ is the (finite or infinite) domain of the dimension $d \in \mathbb{D}$ containing either numerical or categorical values. A k-dimensional subspace $\mathbb{S}[d_1, \ldots, d_k]$ is the Cartesian product of the domains of dimensions $d_1, \ldots, d_k$, given as $dom(d_1) \times \cdots \times dom(d_k)$. If $k = |\mathbb{D}|$, then the subspace is also called a full-space. Note that $\mathbb{S}^*[D]$ extends $\mathbb{S}[D]$ by extending the domain of each dimension with a *null* value. A dataset is a subset of the full-space: $S \subseteq \mathbb{S}^*[\mathbb{D}]$ (meaning that a value can be not stated by using *null*). Each element of S is named a data point.

A k-dimensional point $x[d_1, \ldots, d_k]$ is a vector $\langle x_1, \ldots, x_n \rangle$ where the ith component of x is drawn from the domain of d_i. It means that $x[d_1, \ldots, d_k] \in \mathbb{S}[d_1, \ldots, d_k]$. When the k-dimensional subspace is clear, $\mathbb{S}$ simply denotes $\mathbb{S}[d_1, \ldots, d_k]$ and x simply denotes $x[d_1, \ldots, d_k]$. When $E \subseteq D \subseteq \mathbb{D}$, $x[D][E]$ (or simply $x[E]$) is the projection of $x[D]$ on E.

3.2 Neighborhood Pattern Sampling Problem In traditional discrete data, frequent pattern mining is an extremely popular task due to the support measure. This interestingness measure is intuitive for experts and it is an essential atomic element to build many other interestingness measures. For all these reasons, we adapt the notion of support by considering a pattern as a neighborhood. Instead of considering an exact matching of the dimensional point $x[D]$ with the data point s (i.e., $x[D] = s[D]$), we tolerate a certain distance between $x[D]$ and $s[D]$ by using a p-norm. Given $p \geq 1$, $\|x\|_p$ denotes the p-norm of a k-dimensional point x and is defined as $\left(\sum_{i=1}^{k} |x_i|^p\right)^{1/p}$. For $p = 1$, we get the Manhattan norm; for $p = 2$, we get the Euclidean norm and for p approaching ∞, we get the infinity norm (i.e., $\|x\|_\infty = \max(x_1, \ldots, x_k)$). Let us recall that $\|x - y\|_p$ is a metric and $\|x - y\|_p \leq r$ means that y is near to x for the p-norm and the radius $r \geq 0$. For dealing with a categorical dimension d, we define the absolute difference of two values $x_i, y_i \in dom(d)$ as 0 if $x_i = y_i$, or infinite otherwise[1]. In the same way, for null values, the absolute difference of a value $x_i \in dom(d)$ and *null* is 0 if $x_i = null$, or infinite otherwise.

Given $p \geq 1$ and $r \geq 0$, the neighborhood pattern $x[D]$ is the ball in $\mathbb{S}[D]$ centered around the dimensional point $x[D]$ with a radius r considering the p-norm. As the p-norm and the radius r does not vary within an

[1]For simplicity, we do not discuss other possible approaches because our goal is primarily to address numerical data.

extraction, they will often be omitted in the remainder of the paper and a neighborhood pattern simply corresponds to its center. The full-space language $\mathcal{L}(\mathbb{D})$ is the set of all neighborhood patterns: $\mathcal{L}(\mathbb{D}) = \{x[D] : x \in \mathbb{S} \wedge D \subseteq \mathbb{D}\}$ (note that neighborhood patterns do not contain null values). The dataset language $\mathcal{L}(S)$ is the set of all neighborhood patterns occurring in at least one data point: $\mathcal{L}(S) = \{s[D] \in \mathcal{L}(\mathbb{D}) : s \in S \wedge D \subseteq \mathbb{D}\}$. Figure 1 (left-hand side) illustrates the impact of the norm in two dimensions. It is clear that a p-norm is looser than a q-norm when $p > q$.

Now it is possible to extend the notion of support to neighborhood patterns in $\mathcal{L}(\mathbb{D})$. The *set of neighbors* for a neighborhood pattern $x[D]$ is the set of the data points which projection on D is near to $x[D]$ (for the p-norm and the radius r):

$$n_{p,r}(x[D], S) = \{y \in S \ : \ \|x[D] - y[D]\|_p \leq r\}$$

If two neighborhood patterns x and y contain the same number of neighbors, but the volume of x is twice as small as that of y, it is clear that x is more interesting because it concentrates more data points in a smaller volume. To take into account the volume, we introduce the notion of density. The *density* of the neighborhood pattern $x[D]$ is the number of data points which projection on D is near to $x[D]$ normalized by its volume (for the p-norm and the radius r):

$$d_{p,r}(x[D], S) = \frac{|n_{p,r}(x[D], S)|}{\mathcal{V}_p^{|D|}(r)}$$

where $\mathcal{V}_p^{|D|}(r)$ denotes the volume of the $|D|$-ball of radius r, denoted $\mathcal{B}_p^{|D|}(r)$. Unlike the support, the density is not antimonotone due to the volume definition: $\mathcal{V}_p^{|D|}(r) = (2\Gamma(\frac{1}{p}+1)r)^{|D|}/\Gamma(\frac{|D|}{p}+1)$ where Γ is Euler's gamma function.

Now we introduce the problem that is addressed in the remainder of the paper:

Given a dataset $S \subseteq \mathbb{S}^*[\mathbb{D}]$, a p-norm and a radius $r \geq 0$, the dense neighborhood pattern sampling problem consists in returning a random neighborhood pattern $x[D] \sim d_{p,r}(\mathcal{L}(\mathbb{D}), S)$.

4 Neighborhood Pattern Sampling Algorithm

4.1 Three-Step Random Procedure The intuition at the core of frequent itemset sampling in [4] remains relevant for our pattern sampling problem on numerical data. *A dimensional point close to a random data point is likely to be closed to many data points altogether.* This intuition leads to formulate the first two steps of our non-enumerative sampling procedure:

1. Select randomly a data point x in the dataset with a probability proportional to the size of the powerset of its non-null dimensions.
2. Select a uniformly sampled set E of non-null dimensions of x and return $x[E]$.

Algorithm 1 Dense Neighborhood Pattern Sampling

Input: A dataset $S \subseteq \mathbb{S}^*[\mathbb{D}]$, $p \geq 1$, a radius $r \geq 0$
Output: A random neighborhood pattern drawn according to density

1: Draw a data point $x[\mathbb{D}] \sim w(S)$ where $w(s) = 2^l$ (l is the number of non-null values) for all data points $s \in S$
2: Draw a set of dimensions $E \sim u(2^D)$ where $D \subseteq \mathbb{D}$ are the non-null dimensions of x
3: Draw a dimensional point $z[E] \sim u(\{y \in \mathbb{S}[E] : \|x[E] - y\|_p \leq r\})$
4: **return** $z[E]$

Note that the probability stemming from the non-null dimensions of x in the first step is essential for not introducing a biased towards dimensions occurring in data points having null dimensions. Besides, it is clear that the radius r has no impact on this method.

At this stage, as the neighborhood pattern $x[E]$ belongs to the dataset language, this two-step method is limited to sample the dataset language $\mathcal{L}(S)$ (and not $\mathcal{L}(\mathbb{D})$!). Indeed, the proposed method does not exactly follow the intuition given above: *a dimensional point close to a random data point*. Instead of really drawing a pattern *close to* the data point, the second step only selects a projection of the random data point on E (as if the radius of the neighborhood was zero). So, instead of returning $x[E]$ directly at step 2, we need to add a third and essential step for completing the method:

3. Select a uniformly sampled dimensional point $z[E]$ in the neighborhood of $x[E]$ and return $z[E]$.

Thanks to this new step, our three-step random procedure considers all dimensional points of the full-space language $\mathcal{L}(\mathbb{D})$. In particular, it is easy to see that all dimensional points for which the density is zero are not drawn because they do not belong to any neighborhood of a data point. Algorithm 1 sketches this exact and non-enumerative random procedure. As main theoretical result, Theorem 4.1 proves the correctness of this algorithm:

THEOREM 4.1. *Given a dataset $S \subseteq \mathbb{S}^*[\mathbb{D}]$, Algorithm 1 generates a neighborhood pattern $z[E]$ according to a probability density function proportional to its density: $z[E] \sim density_{p,r}(\mathcal{L}(\mathbb{D}), S)$.*

Proof. Due to the lack of space, we demonstrate this result by considering that all dimensions $\mathbb{D}$ are numerical.

We are going to prove that the probability of selecting a point in the ball centered on $z[E]$ with a radius $\epsilon \ll r$, denoted by $\mathcal{B}_p(z[E], \epsilon)$, is proportional to the density of $z[E]$ times the volume of this ball:

$$\mathbb{P}(y \in \mathcal{B}_p(z[E], \epsilon)) = \frac{d_{p,r}(z[E], S)}{Z} \times \mathcal{V}_p^{|E|}(\epsilon)$$

where $Z = \sum_{D \subseteq \mathbb{D}} \int_{\mathbb{S}[D]} |n_{p,r}(x[D], S)| / \mathcal{V}_p^{|D|}(r) \mathrm{d}x$. The normalizing constant Z can be rewritten as the sum of the volumes on each subspace:

$$Z = \sum_{D \subseteq \mathbb{D}} \sum_{s \in S_{/D}} \underbrace{\int_{\mathcal{B}_p^{|D|}(r)} 1/\mathcal{V}_p^{|D|}(r) \mathrm{d}x}_{=1} = \sum_{D \subseteq \mathbb{D}} |S_{/D}| = \sum_{s \in S} 2^{D(s)}$$

where $\mathcal{B}_p^{|D|}(r)$ is a $|D|$-ball of radius r, $S_{/D}$ is the set of data points in S having no null value on D and $D(x)$ gives the number of non-null dimensions for x.

In order to compute $\mathbb{P}(y \in \mathcal{B}_p(z[E], \epsilon))$, we have to marginalize out x (the data point drawn at step 1) and F (the set of dimensions drawn at step 2):

$$\mathbb{P}(y \in \mathcal{B}_p(z[E], \epsilon)) = \sum_{x \in S, F \subseteq D(x)} \mathbb{P}(x) \times \mathbb{P}(F/x) \times \mathbb{P}(y \in \mathcal{B}_p(z[E], \epsilon)/x, F)$$

Considering Algorithm 1, it is easy to see that the probability to select x in S using step 1 is: $\mathbb{P}(x) = 2^{D(x)} / \sum_{s \in S} 2^{D(s)}$. Then, the probability to select a subset F of dimensions in $D(x)$ using step 2 is: $\mathbb{P}(F/x) = 1/2^{D(x)}$. Finally, the probability $\mathbb{P}(y \in \mathcal{B}_p(z[E], \epsilon)/x, F)$ is equal to zero if $x \notin n_{p,r}(z[E], S)$ or $F \neq E$. Otherwise, as $z[E]$ is uniformly sampled into the neighborhood of $x[E]$, we have: $\mathbb{P}(y \in \mathcal{B}_p(z[E], \epsilon)/x, E) = \int_{\mathcal{B}_p(z[E], \epsilon)} \frac{\mathrm{d}y}{\mathcal{V}_p^{|E|}(r)} = \frac{\mathcal{V}_p^{|E|}(\epsilon)}{\mathcal{V}_p^{|E|}(r)}$.

Using the above three probabilities, we obtain:

$$\begin{aligned}
\mathbb{P}(y \in \mathcal{B}_p(z[E], \epsilon)) &= \sum_{x \in n_{p,r}(z[E], S)} \frac{2^{D(x)}}{\sum_{s \in S} 2^{D(s)}} \times \frac{1}{2^{D(x)}} \times \frac{\mathcal{V}_p^{|E|}(\epsilon)}{\mathcal{V}_p^{|E|}(r)} \\
&= \sum_{x \in n_{p,r}(z[E], S)} \frac{1}{\sum_{s \in S} 2^{D(s)}} \times \frac{\mathcal{V}_p^{|E|}(\epsilon)}{\mathcal{V}_p^{|E|}(r)} \\
&= \underbrace{\frac{|n_{p,r}(z[E], S)|}{\mathcal{V}_p^{|E|}(r)}}_{d_{p,r}(z[E], S)} \times \underbrace{\frac{1}{\sum_{s \in S} 2^{D(s)}}}_{1/Z} \times \mathcal{V}_p^{|E|}(\epsilon)
\end{aligned}$$

□

Each of the first two steps (Algorithm 1) achieves a uniform sampling on a finite set of elements (the data points S or the powerset of non-null dimensions of x) and are rather simple to implement. In contrast, the third step addresses an infinite set of points by considering the neighborhood of the dimensional point $x[E]$. This step is semantically interesting because it brings diversity. Because it is non-trivial, next section investigates how it can be performed efficiently.

4.2 Third Step The draw of a dimensional point $z[E]$ in the neighborhood of $x[E]$ is equivalent to the uniform sampling of a point inside the p-norm $|E|$-ball of radius r (plus a translation of x). In the case of the ∞-norm, this draw consists in uniformly drawing a point in the hypercube centered on the origin having its edges of length $2 \times r$, parallel to the axes. This can easily be achieved by sampling each component z_i uniformly in the interval $[-r, +r]$. For the 1-norm or the 2-norm, the draw is more complex. A naive method would be to perform a rejection sampling by drawing a point in the hypercube (as above) and by rejecting this point whenever its distance from the origin is greater than r for the 1-norm (or the 2-norm). This approach is inefficient when the number of dimensions in E increases due to the curse of dimensionality. For instance, the rejection probability is greater than 0.99 as soon as a 5-dimensional (resp. 9-dimensional) space is considered with the 1-norm (resp. the 2-norm). This approach is not feasible in practice when datasets have dozens of dimensions (see Section 5).

We present two specific and efficient algorithms for uniformly sampling the p-norm $|E|$-ball of radius r for $p = 1$ and $p = 2$. The first algorithm dedicated to the 1-norm (see Algorithm 2) is mainly based on selecting a point from a unit simplex, uniformly at random [26]. As a reminder, any point inside the simplex has the sum of its components less than 1. After, the components of such a sampled point are scaled according to the radius r and randomly opposite (line 6). Algorithm 2 returns a k-dimensional point uniformly drawn inside the ball $\mathcal{B}_1^k(r)$ in $O(k \ln k)$ time due to the sort of components.

The second algorithm (see Algorithm 3) reformulates the technique used in [25] that returns points uniformly distributed on the unit k-sphere. It rests on a property of the normal distribution (the k-dimensional canonical normal density function has constant probability on the surfaces of k-dimensional spheres with common centers). Note that we use the Box-Muller transform [5] for the generation of a normal distribution. Algorithm 3 returns a k-dimensional point uniformly drawn inside the ball $\mathcal{B}_2^k(r)$ in $O(k)$ time.

4.3 Global Complexity Analysis A single pass over the dataset is necessary for preparing the first step of Algorithm 1 by computing the weight w for

Algorithm 2 Uniform 1-norm Ball Sampling

Input: A dimension number k, a radius r
Output: A k-dimensional point uniformly drawn inside the ball $\mathcal{B}_1^k(r)$

1: Draw a k-dimensional point y according to the k-dimensional uniform distribution
2: Sort the k components of y in ascending order
3: $p := 0$
4: **for** $i = 1$ **to** k **do**
5: Draw $u \in [0, 1]$ according to the uniform distribution
6: $x_i := r \times (y_i - p) \times \textbf{signum}(u - 0.5)$
7: $p := y_i$
8: **end for**
9: Permute randomly components of x
10: **return** x

Algorithm 3 Uniform 2-norm Ball Sampling

Input: A dimension number k, a radius r
Output: A k-dimensional point uniformly drawn inside the ball $\mathcal{B}_2^k(r)$

1: Draw a k-dimensional point x according to the k-dimensional standard normal distribution
2: Draw a value $u \in [0, 1]$ according to the uniform distribution
3: $x := r \times u^{1/k} \times \frac{x}{\|x\|_2}$
4: **return** x

all data points. After, the draw of a neighborhood pattern requires to access the sampled data point (in time $O(\ln |S|)$) and to select a subset of dimensions (in time $O(|\mathbb{D}|)$). Finally, the time complexity varies according to the considered p-norm at step 3 (proofs of properties are omitted due to lack of space):

PROPERTY 4.1. *Given a dataset $S \subseteq \mathbb{S}^*[\mathbb{D}]$, a family of k realizations of a random neighborhood pattern $x \sim density_{p,r}(\mathcal{L}(\mathbb{D}), S)$ can be generated in time $O(|S| \times |\mathbb{D}| + k(|\mathbb{D}| + \ln |S|))$ for the 2-norm and the ∞-norm and in time $O(|S| \times |\mathbb{D}| + k(|\mathbb{D}| \ln |\mathbb{D}| + \ln |S|))$ for the 1-norm.*

Only the use of the 1-norm leads to an harder complexity of the three-step method compared to the two-step method. In practice, the average pattern draw time in the datasets used in the next section does not exceed a few tens of milliseconds (whatever the p-norm), except for the letter dataset where drawing a pattern requires on average 360 milliseconds.

5 Experimental Study

This experimental study aims at evaluating the quality of neighborhood patterns returned by the three-step random procedure. It is always difficult to show that extracted patterns are relevant since pattern mining is an unsupervised task. In Section 5.1, we assess the sampled patterns via a swap randomization protocol inspired from [13]. As an objective evaluation metric in Section 5.2, we will also measure the accuracy of classifiers built from the sampled neighborhood patterns as done in [4].

Experiments are conducted on 19 datasets coming from the UCI ML repository (`archive.ics.uci.edu/ml`). We normalize numerical data using z-score: $z = (x - \mu)/\sigma$ where μ is the mean of the population and σ is the standard deviation of the population. All experiments are performed on a 2.5 GHz Xeon processor with the Linux operating system and 2 GB of RAM memory. Algorithms are implemented in Java and the source code is available at `www.info.univ-tours.fr/~soulet/prototype/sdm18/`.

5.1 Plausibility and Diversity of Sampled Neighborhood Patterns Our first goal is to assess the significance of neighborhood pattern sampling with p-norms (for $p \in \{1, 2, \infty\}$) using density on the full-space language $\mathcal{L}(\mathbb{D})$. For this purpose, we compare this sampling approach with three others:

- **Neighborhood patterns on $\mathcal{L}(S)$:** We remove the third step for restraining the sampling to $\mathcal{L}(S)$ (see Section 4.1).

- **Interval pattern:** MinIntChange algorithm [19] for mining a condensed representation of interval patterns is not sufficiently scalable for dealing with UCI datasets. Instead, we have developed interval pattern sampling method simulating 1) a complete extraction of all interval patterns (not only the condensed representation) and 2) a draw of interval patterns proportional to their frequency. More precisely, this approach uses the same first two steps of Algorithm 1 and replaces the third step by an interval generation. For each component x_i of dimension d, this interval generation selects uniformly two values $m_i, M_i \in dom(d)$ such that $x_i \in [m_i, M_i]$.

- **3-bins itemset:** We first discretize the dataset using an equal-frequency discretization method with three bins (i.e., each discrete interval contains the same number of values). Then, we apply our pattern sampling algorithm. On categorical data, it is exactly equivalent to that of [4] because the volume of a neighborhood pattern is 1 when there is no numerical dimension.

We use two measures for assessing the quality of patterns resulting from each method:

	Two-step method in $\mathcal{L}(S)$						Three-step method in $\mathcal{L}(\mathbb{D})$									
	1-norm		2-norm		∞-norm		1-norm		2-norm		∞-norm		Interval		3-bins	
Dataset	Plau.	Div.	Plau.	Div.	Plau.	Div.	Plau.	Div.	Plau.	Div.	Plau.	Div.	Plau.	Div.	Plau.	Div.
abalone	0.79	0.99	0.53	0.99	0.38	0.99	**0.80**	0.99	0.54	0.99	0.29	0.99	0.24	**1.00**	0.74	0.51
adult	**0.31**	0.96	0.27	0.96	0.24	0.96	0.30	0.96	0.25	0.96	0.24	0.96	0.17	**0.97**	0.17	0.95
breast	0.79	0.70	0.52	0.70	0.34	0.70	**0.82**	0.91	0.52	0.81	0.32	**0.94**	0.35	**0.94**	0.64	0.44
bupa	0.13	0.77	0.07	0.76	0.05	0.77	0.13	0.90	0.07	0.87	0.05	0.89	0.04	**0.95**	**0.15**	0.31
crx	**0.38**	0.78	0.34	0.78	0.29	0.78	0.36	**0.79**	0.31	0.78	0.31	0.77	0.24	0.77	0.21	0.59
glass	**0.56**	0.56	0.28	0.56	0.17	0.56	**0.56**	0.74	0.29	0.67	0.16	0.78	0.18	**0.81**	0.38	0.35
heart*	**0.81**	0.64	0.38	0.64	0.25	0.64	0.79	0.71	0.39	0.66	0.23	0.71	0.25	**0.81**	0.31	0.54
hypo	**0.46**	**0.99**	0.34	**0.99**	0.31	0.98	**0.46**	**0.99**	0.34	**0.99**	0.29	**0.99**	0.32	**0.99**	0.28	0.97
ionosphere*	**1.00**	0.55	**1.00**	0.55	0.93	0.55	**1.00**	**0.58**	**1.00**	0.56	0.88	0.42	0.60	0.49	0.07	0.10
iris	0.35	0.10	0.29	0.10	0.25	0.10	0.33	0.36	0.28	0.29	0.21	0.44	0.21	**0.62**	**0.46**	0.01
letter*	**0.95**	0.99	0.55	0.99	0.27	0.99	0.94	**1.00**	0.56	0.99	0.19	**1.00**	0.18	**1.00**	0.10	0.98
new-thyroid	0.20	0.28	0.14	0.28	0.11	0.28	0.20	0.62	0.14	0.54	0.10	0.68	0.11	**0.83**	**0.24**	0.08
pima	**0.31**	0.94	0.14	0.94	0.09	0.94	**0.31**	0.97	0.13	0.96	0.08	0.96	0.08	**0.99**	0.19	0.70
sick	**0.40**	0.98	0.28	0.98	0.23	0.98	0.39	0.98	0.27	0.98	0.22	0.98	0.26	**0.99**	0.21	0.96
spambase*	0.72	**0.83**	0.66	**0.83**	0.26	**0.83**	**0.88**	**0.83**	0.57	**0.83**	0.37	0.81	0.09	**0.83**	0.38	0.66
waveform*	0.97	0.98	0.77	0.98	0.34	0.98	**0.98**	0.98	0.78	0.98	0.37	0.90	0.24	**1.00**	0.02	0.73
wdbc*	**1.00**	0.80	0.98	0.80	0.68	0.80	0.98	**0.81**	0.98	0.80	0.73	0.48	0.46	**0.81**	0.03	0.17
wine	**0.96**	0.73	0.66	0.73	0.42	0.73	0.95	0.78	0.66	0.76	0.38	0.63	0.33	**0.82**	0.41	0.29
yeast	**0.15**	0.92	0.07	0.92	0.05	0.92	**0.15**	0.96	0.07	0.95	0.04	0.96	0.03	**0.97**	0.09	0.54
Average:	0.59	0.76	0.44	0.76	0.30	0.76	**0.60**	0.83	0.43	0.81	0.29	0.80	0.23	**0.87**	0.27	0.52

Table 1: Plausibility and diversity for UCI benchmarks with a radius equal to 1

- **Plausibility:** Plausibility measures whether the mined patterns truly characterize the dataset or whether they result from chance. It is defined as the probability that a sampled pattern $X \in \mathcal{S}$ having an interest m greater than $\delta \in [0, \infty)$ in S has not an interest m greater than δ in a randomized dataset S^*. Note that the randomized dataset S^* shares the same characteristics with S but the values of a same dimension have been permuted. The idea of this randomized dataset is to erase all correlations of S. Thus, all the patterns in S^* are considered as spurious. More formally, we define the plausibility w.r.t. m (e.g., density or support) for a sample $\mathcal{S}$ as follows:

$$\int_0^\infty \frac{|\{m(X,S) \geq \delta \wedge m(X,S^*) < \delta | X \in \mathcal{S}\}|}{|\{m(X,S) \geq \delta | X \in \mathcal{S}\}|} \mathrm{d}\delta$$

  This protocol is inspired from [13] for evaluating the significance of results.

- **Diversity:** We define the diversity of k sampled patterns as the number of distinct equivalent classes divided by k. Two patterns $x[D]$ and $y[E]$ are equivalent when they share the same data points as neighbors i.e., $n_{p,r}(x[D], S) = n_{p,r}(y[E], S)$.

Table 1 reports the plausibility (for $10,000$ patterns) and the diversity (for $100,000$ patterns) for each UCI benchmarks considering a radius equal to 1. Each reported evaluation measure is the arithmetic mean of 10 repeated measurements (interval confidence are narrow enough to be omitted). For each line, the best approach is highlighted in bold.

Overall neighborhood pattern sampling is the best sampling method in terms of plausibility. Only the plausibility using the ∞-norm is not completely satisfactory. In this case, a neighborhood pattern behaves as an interval pattern of side $2r$ and are rather sensitive to noise in large dimensions. Whatever the number of steps, the plausibility of neighborhood pattern sampling with the 1-norm is significantly better than that with the 2-norm. Neighborhood pattern sampling with $p \in \{1, 2\}$ is particularly relevant in datasets with a large number of numerical dimensions. Apart spambase, the 6 datasets having at least 10 numerical dimensions (marked by a star) are also the 6 datasets having a plausibility for 1-norm neighborhood patterns twice more important than that for discretized itemsets. Thus, neighborhood patterns seem to be more resistant to the "curse of dimensionality".

As expected, the diversity of interval patterns is the highest while that of discretized patterns is the worst (reflecting the inherent information loss). Neighborhood patterns have good diversity since for the 1-norm three-step method, the gap with the interval patterns is a few percent. Importantly, the third step increases the diversity of neighborhood patterns showing the true interest to consider the full-space language $\mathcal{L}(\mathbb{D})$.

Figure 2 presents the averages of evaluation measures (plausibility with $10,000$ patterns and diversity with $100,000$ patterns) for the 19 UCI benchmarks when the radius varies from 0 to 4. Neighborhood pattern sampling has clearly the best plausibility when the radius is low. For radii below 1, the plausibility of the neighborhood patterns is greater than that for the intervals. About the diversity, we remark that increasing

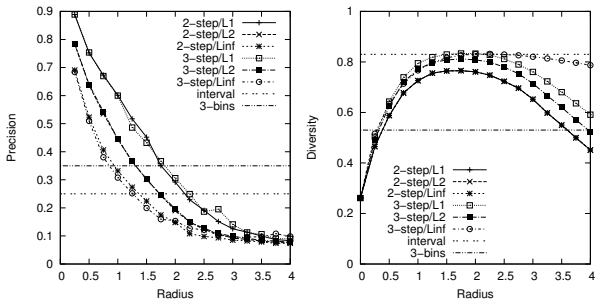

Figure 2: Plausibility and diversity of neighborhood pattern sampling with radius

	Sampling-based classifiers				Baseline classifiers		
Dataset	1	2	∞	3-bins	CBA	CMAR	CPAR
crx	86.4	86.5	86.7	86.7	84.7	84.9	85.7
glass	70.5	72.9	72.4	67.3	73.9	70.1	74.4
heart*	81.9	82.6	82.6	78.1	81.9	82.2	82.2
hypo	97.5	97.4	97.5	95.3	98.9	98.4	98.1
iono.*	83.2	83.2	84.6	74.4	92.3	91.5	92.6
iris	96.7	96.7	96.0	96.7	94.7	94.0	94.7
pima	76.8	76.0	76.7	69.0	72.9	75.1	73.8
sick	96.8	96.6	96.7	93.5	97.0	97.5	96.8
wave.*	82.6	82.1	81.2	72.3	80.0	83.2	80.9
wine	94.4	96.6	95.5	92.2	95.0	95.0	95.5
avg:	86.7	87.1	87.0	82.5	87.1	87.2	87.5

Table 2: Pattern-based classification based on neighborhood pattern sampling

the radius increases the diversity until a certain level (depending on the method). For the three-step sampling, the diversity is higher than that of itemsets for radii less than 4. The diversity of neighborhood pattern sampling with the ∞-norm even managed to reach that of interval patterns.

In summary, it is clear that the three-step method with the 1-norm is the best compromise between plausibility and diversity. It is also interesting to use the Euclidean norm which can be more intuitive on certain datasets (e.g., spatial data). In all situations, the addition of the third step increases the diversity with a marginal plausibility loss in the worst case. A radius around 1 provides a good plausibility and a satisfactory diversity when the z-score is used as standardization.

5.2 Accuracy of Sampling-Based Classification

In this section, we evaluate the interest of neighborhood patterns in the context of pattern-based classification. Our goal is to apply a CBA-like classification [22] starting from a sample of neighborhood patterns to measure whether the accuracy is comparable to traditional associative classifiers which are based on a complete exploration of the search space. In a nutshell, it consists in building an associative classifier based on a sample $\mathcal{S}$ of 10,000 neighborhood patterns. For each pattern $x[D] \in \mathcal{S}$, an association rule $x[D] \to c$ is derived iff $x[D] \to c$ has a confidence greater than 0.5 (here, $conf(X \to c) = n_{p,r}(x[D], S_c)/n_{p,r}(x[D], S)$ where the subdataset S_c contains all the data points of class c). The CBA approach is used for making prediction. Given a new data point y, the rule $x[D] \to c$ such that $\|x[D] - y[D]\|_p \leq r$ and that maximizes the confidence (and if necessary, the neighborhood) is applied to predict the class c. For each dataset S and each norm p, an optimal radius r is found by means of a cross-validation on the training dataset where the accuracy is optimized. Table 2 reports the accuracy of this sampling-based classification for the three norms. We apply this same approach with a sample of 10,000 frequent itemsets on a discretized dataset (see the fifth column of Table 2). Finally, we compare our approach with three associative classifiers (i.e., CBA, CMAR and CPAR) as baseline in the three last columns. We report the accuracy results given in [29] for the datasets in common with the previous section.

The first observation is that the three norms have a very similar behavior (average accuracy between 86.7% and 87.1%). Indeed, the adjustment of the radius (by means of cross-validation) makes it possible to find a good compromise between plausibility and diversity which attenuates the impact of each norm whatever the dataset. Interestingly, the use of neighborhood patterns directly on numerical data improves the accuracy with respect to a similar sampling approach on the binarized data. In particular, neighborhood patterns are more relevant for datasets with a large number of numerical dimensions (e.g., ionosphere or waveform).

The most important observation is that our CBA-like approach based on neighborhood pattern sampling is as accurate as associative classifiers of the literature. It means that the sample of neighborhood patterns is sufficiently interesting and representative of the entire search space. Obviously this result is all the more interesting as unlike the complete methods, the set of neighborhood patterns is sampled in a few seconds.

6 Conclusion

We introduced a new pattern mining method in numerical data that abandons the paradigm of the complete enumeration to that of an instant access to the pattern language. An originality of our work is the proposal of neighborhood pattern which is a pattern structure that does not separately consider each dimension due to the use of a metric. The experimental study shows that neighborhood patterns have a high precision while maintaining excellent diversity in comparison with previous literature approaches. In the context of

associative classification, a sample of neighborhood patterns gives an accuracy comparable to the traditional approaches traversing the entire search space. Despite the infinite number of neighborhood patterns, a new method was proposed to sample according to density without using a stochastic process. After a preliminary pass over the data, this three-step method is effective enough to instantly return patterns even on large datasets.

Our work goes in the direction of the interactive data exploration using pattern mining [12] that encourages a tight coupling between the user and the mining system. Although it focused on the density measure, we would like to extend this technique to other interestingness measures that may include user feedback. Rather than immediate use by an end-user, we also intend to benefit from this method of pattern sampling in numerical data as an elementary block for subspace clustering [21] without considering discretization.

References

[1] E. Aksehirli, B. Goethals, E. Muller, and J. Vreeken. Cartification: A neighborhood preserving transformation for mining high dimensional data. In *ICDM*, pages 937–942. IEEE, 2013.

[2] S. D. Bay. Multivariate discretization for set mining. *Know. and Inf. Syst.*, 3(4):491–512, 2001.

[3] M. Boley, T. Gärtner, and H. Grosskreutz. Formal concept sampling for counting and threshold-free local pattern mining. In *SDM*, pages 177–188. SIAM, 2010.

[4] M. Boley, C. Lucchese, D. Paurat, and T. Gärtner. Direct local pattern sampling by efficient two-step random procedures. In *KDD*, pages 582–590, 2011.

[5] G. E. Box and M. E. Muller. A note on the generation of random normal deviates. *The annals of mathematical statistics*, (29):610–611, 1958.

[6] T. Calders, B. Goethals, and S. Jaroszewicz. Mining rank-correlated sets of numerical attributes. In *KDD*, pages 96–105. ACM, 2006.

[7] L. Di-Jorio, A. Laurent, and M. Teisseire. Mining frequent gradual itemsets from large databases. In *IDA*, pages 297–308. Springer, 2009.

[8] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *ICML*, pages 194–202, 1995.

[9] V. Dzyuba, M. van Leeuwen, and L. De Raedt. Flexible constrained sampling with guarantees for pattern mining. *Data Mining and Know. Disc.*, pages 1–28, 2017.

[10] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining with optimized two-dimensional association rules. *ACM Trans. Database Syst.*, 26(2):179–213, 2001.

[11] A. Giacometti and A. Soulet. Frequent pattern outlier detection without exhaustive mining. In *PAKDD*, pages 196–207. Springer, 2016.

[12] A. Giacometti and A. Soulet. Interactive pattern sampling for characterizing unlabeled data. In *IDA*, pages 99–111, 2017.

[13] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas. Assessing data mining results via swap randomization. *ACM Trans. on Know. Disc. from Data*, 1(3):14, 2007.

[14] H. Grosskreutz and S. Rüping. On subgroup discovery in numerical domains. *Data mining and knowledge discovery*, 19(2):210–226, 2009.

[15] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.

[16] D. J. Hand. Pattern detection and discovery. *Pattern Detection and Discovery*, 2447:1–12, 2002.

[17] M. A. Hasan and M. J. Zaki. Output space sampling for graph patterns. *PVLDB*, 2(1):730–741, 2009.

[18] S. Jaroszewicz and M. Korzeń. Approximating representations for large numerical databases. In *SDM*, pages 521–526. SIAM, 2007.

[19] M. Kaytoue, S. O. Kuznetsov, and A. Napoli. Revisiting numerical pattern mining with formal concept analysis. In *IJCAI*, pages 1342–1347, 2011.

[20] R. M. Konijn, W. Duivesteijn, W. Kowalczyk, and A. Knobbe. Discovering local subgroups, with an application to fraud detection. In *PAKDD*, pages 1–12. Springer, 2013.

[21] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. on Know. Disc. from Data*, 3(1):1, 2009.

[22] B. Ma, W. Liu, Y. Hsu, and W. Liu. Integrating classification and association rule mining. In *KDD*, 1998.

[23] S. Moens and M. Boley. Instant exceptional model mining using weighted controlled pattern sampling. In *IDA*, pages 203–214, 2014.

[24] S. Moens and B. Goethals. Randomly sampling maximal itemsets. In *Proc. of the KDD Workshop on IDEA*, pages 79–86. ACM, 2013.

[25] M. E. Muller. A note on a method for generating points uniformly on n-dimensional spheres. *Communications of the ACM*, 2(4):19–20, 1959.

[26] N. A. Smith and R. W. Tromble. Sampling uniformly from the unit simplex. *Johns Hopkins University, Tech. Rep*, pages 1–6, 2004.

[27] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 1–12, 1996.

[28] N. Tatti. Itemsets for real-valued datasets. In *ICDM*, pages 717–726. IEEE, 2013.

[29] X. Yin and J. Han. CPAR: Classification based on predictive association rules. In *SDM*, pages 331–335, 2003.