Local Pattern Detection and Clustering
Are there substantive differences?

Frank Höppner

University of Applied Sciences Braunschweig/Wolfenbüttel
Robert Koch Platz 10-14
D-38440 Wolfsburg, Germany

Abstract. The starting point of this work is the definition of local pattern detection given in [10] as the unsupervised detection of local regions with anomalously high data density, which represent real underlying phenomena. We discuss some aspects of this definition and examine the differences between clustering and pattern detection (if any), before we investigate how to utilize clustering algorithms for pattern detection. A modification of an existing clustering algorithm is proposed to identify local patterns that are flagged as being significant according to a statistical test.

1 Introduction

Knowledge discovery in databases (KDD) aims at detecting valid, novel, potentially useful, and ultimately understandable patterns in data [8]. Many tools in KDD aim at a global characterization of the data, such as decision trees or clustering partitions. The more recent technique of association rule mining, however, investigates into more local phenomena that do not characterize the database as a whole but only a small subpopulation. Usually, association rule mining is considered as the most prominent approach to *local pattern detection*. However, experiments with (standard) association rule mining are often somewhat frustrating, because the number of local patterns often becomes that large that it is no longer manageable. And even worse, most of these patterns – flagged as being potentially interesting – turn out to be neither useful nor valid in the application context. For a deeper discussion see [4]. The definition of local pattern detection given by Hand [10] takes these aspects into account. The main points in his definition are:

1. A local pattern is a data vector serving to describe an *anomalously high local density of data points* when compared to a background model:

$$\text{data} = \text{background\_model} + \text{pattern} + \text{random\_component} \quad (1)$$

[1] F. Höppner: Local Pattern Detection and Clustering – Are there substantive differences. In: *Local Pattern Detection*, LNAI 3539, 53–70, © Springer 2005.

2. Local pattern detection is unsupervised in the sense that no information but the data itself is given to find out what patterns may be present in the database, if any.
3. Local pattern detection is about inferring from observations, therefore patterns must represent real phenomena and not just noise.

In this paper we will contrast the goals of local pattern detection with those of clustering (section 2) and discuss some potential problems and consequences when following the definition above (section 3). Whether a flagged pattern is substantive or not is influenced by two different facts, one is the statistical significance of an identified candidate patterns, the other is the robustness of the applied algorithms, that is, the sensitivity to initial parameters, which is a problem with many clustering algorithms in particular. We will discuss consequences and candidate algorithms in section 4. In section 5 we will finally discuss a pattern detection algorithm that has many of the desired properties discussed before, which will be illustrated via some examples in section 6.

2 Local Pattern Detection vs. Clustering

At first glance, the before-mentioned description of pattern detection sounds almost identical to clustering. Here is an exemplary definition from the literature:

> "Clusters may be described as connected regions of multi-dimensional space containing a relatively high density of points, separated from other such regions by a region containing a relatively low density of points" [7]

The identification of (local) regions with high data density (point 1 in the definition) and the fact that pattern detection is an unsupervised approach (point 2) establishes a strong relationship between local pattern detection and clustering.

In accordance with point 1 of the definition, we could compose our data model out of several Gaussian distributions and a single uniform distribution. If we think of the uniform distribution as the background model in (1) and the Gaussian distributions as the patterns, the differences between pattern detection and clustering begin to blur. Standard mixture decomposition could be applied to identify the parameters of the models – and if the parameters of the Gaussian indicate that only a small portion of the input space is affected (small covariance), we could speak of an identified pattern.

May be it is surprising that traditional definitions of clustering [7, 12, 14] do not contain anything similar to the third statement in the definition of local pattern detection, which refers to the statistical "validity" of identified clusters.[1] While it is not mentioned in the definitions, the problem that "the resulting

[1] With some clustering algorithms, e.g. when the number of clusters has to be fixed in advance, so called *validity measures* are used to "validate" the results. Even if these measures are not purely heuristic in nature but investigate statistical properties of a partition, they seldomly take the role of a statistical test.

clustering procedures have no known significant theoretical properties" [5] is well recognized. But unfortunately not much has changed since Hartigan stated in 1975 [12] that clustering algorithms "are not yet an accepted inhabitant of the statistical world". This makes the current position in pattern detection even more similar to that in clustering, because in both fields some theoretical framework is missing. (Given that Hartigan made his statement in 1975 and also given the lack of progress in this concern, the "development of a theoretical base" [10] for pattern detection appears really challenging.)

Rather than by using statistical tests, in machine learning overfitting is often avoided by employing a regularization framework. In contrast to statistics, such a framework aims at limiting the variability of the models, but does not care primarily about the statistical significance of the result. On the other hand, if the assumptions of the used statistical model (which are always present) are violated (which may happen quite easily in KDD) there is not much left that distinguishes regularization from statistical relevance tests.

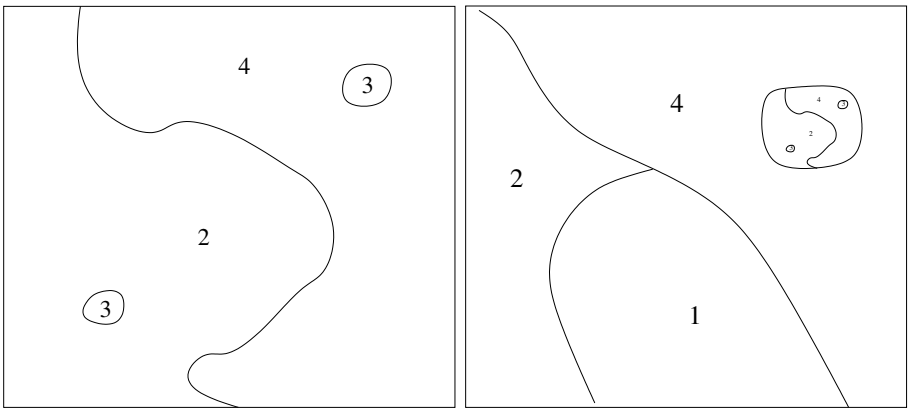
Up to this point the reader may agree that clustering and (today's) local pattern detection are indeed very similar. The only distinction that is left is the explicit focus on *local* patterns, which cannot be found in clustering. We will see in the following, however, that this is not enough for a substantive distinction. So a provocative definition of local pattern detection could be "clustering, done right".

3 What is the background model?

Having a background model defining the normal situation enables us to apply a statistical test to see whether some observations deviate significantly from the background model or not. Thus the background model plays a key role in detecting substantive local patterns. On closer inspection, however, it becomes clear that this works well only under the assumption that the background model is valid. And determining the validity of the background model may be as difficult as determining the validity of a cluster (or pattern) without a supporting background model.

This leads us to the general question of how to select a background model. A good candidate for a background model, when little domain knowledge is available, might be the uniform distribution. Figure 1(a) illustrates a hypothetical data set. On the right hand side the data density is 4.0 (per some area) and on the left it is 2.0, both sides are occupied by approximately the same number of data objects. On both sides, there are smaller regions in which the data density is 3.0; intuitively, these are the "patterns". If we assume a uniform distribution as the background model, we obtain an average density of 3.0, which perfectly corresponds to the density of our patterns. Therefore, *this* background model would not flag them as substantive patterns. The background model may flag the larger regions as deviations from the background model, but they do not qualify as patterns due to their size. (By the way, does the small cluster on the

right qualify as a pattern? It represents a deviation from the background model, but its data density is smaller rather than larger.)



(a) The numbers in the areas denote the average data density per volume.

(b) Locality is a matter of scale: The environment of figure.

Fig. 1. A hypothetical data set.

The point in Fig. 1 is of course that a single, simple background model will not work. Either the background model must be flexible and complicated, or it must be possible to define different background models in different parts of the data space.[2] In the context of clustering, we could say that we have two clusters in Fig. 1(a), the left and the right part of the figure. And within each cluster, there is a small subcluster – which we may call a local pattern. But indeed, we never know whether we currently observe a cluster, a background model or a local pattern, unless we know about the scale at which we look on the data. (The whole Fig. 1(a) may be a local pattern itself – in the upper right corner of the coarser view of figure 1(b).) Even though we may be interested in local patterns only, we *have to* carefully consider structures at any larger scale. In analogy to (1) we could try to express this fact by a recursive definition

$$\begin{aligned} \text{data} &= \text{model} + \text{noise} \qquad \text{where} \\ \text{model} &= \text{atomic\_model} \mid \text{background\_model} + \text{model}^* \end{aligned} \tag{2}$$

[2] In association rule mining, the minimum support threshold may be seen as being part of a very coarse background model. In a k-dimensional boolean space, we have 2^k possible configurations. For n records and a uniform distribution, we expect $\frac{n}{2^k}$ objects per combination. The $\min_{supp}$ threshold, however, is the same for *all item sets of any size* and does not depend on k.

where model* means that any number of models can supplement the background model. Thus, the data may be represented by a hierarchical tree of models, where the same model may serve as a cluster in one level and a background model in another. Local patterns (in the sense of small clusters) can be considered as the leaves of this model tree.

A background model helps with the identification of substantive local patterns only if the background model itself is valid. Simple examples (fig. 1(a)) show that we cannot restrict ourselves to a global, simple background model. Estimating a valid background model of arbitrary complexity (eq. (1)) in one step seems unrealistic. Utilizing erroneous or inadequate background models puts the validity of the identified local patterns in question. The most promising approach is to start with a simple model (whose parameters can be estimated easily and robustly) and use this as the basis for the next hierarchy level to come (eq. 2)). Then, for the identification of all models in later stages, we already benefit from the existence of a valid background model (stepwise refinement). This approach allows us to stick to simple background models, such as the uniform distribution, even in cases like figure 1(a) (only the boundary remains to be determined). And this approach also underlines that we cannot focus on local structure only, but must carefully investigate structures at any scale.

4 Escaping from Heuristic Thresholds

One can easily find clustering algorithms that respect these ideas. For instance, we could start by estimating the parameters of a mixture of Gaussians. In [5] (p. 558) a statistical test is proposed to decide whether a cluster may have been generated from a single Gaussian. Such tests can be applied to each cluster for validation; if the test fails, we may generate a new data set that contains only the data objects belonging to this cluster (for instance, we could sample from the original data set using the a posteriori probabilities of being generated by the Gaussian). We then apply the same clustering algorithm once more, which leads us to a hierarchical subdivision of the previously discovered cluster. The data set refinement is stopped if such a refinement cannot be justified by the data any longer. A similar method (where the tests are not statistical in nature) can be found in [9].

When implementing such algorithms technical details become highly relevant for failure or success, such as:

- Did the algorithm yield the correct solution (that is, did the expectation maximization algorithm yield the (globally) optimal solution)?
- Was the assumption of Gaussian distributions justified?
- If we need data density estimations (e.g. to detect the clusters in Fig. 1(a)), did we select an appropriate size of the area that is used to estimate the density?

Most clustering algorithms require a couple of initialization parameters – and are generally more sensitive to their setting than we would like them to be. The

(background) models are not the only information we are processing, and the same effort to validate background models and patterns should also be spent on any other step in the line of processing, because invalid intermediate results also deteriorate the correctness of our final patterns. The theoretical advantage of using statistical tests with the background model is worth nothing if the algorithms pass ill-formed pattern candidates to the test.

With every (heuristic) threshold an algorithm requires we increase the risk of processing unvalidated data. And a lot of decisions may be necessary, in particular in the preprocessing phase. Most often, the parameters are chosen on the basis of some small sample and visual inspection, but in KDD we cannot be sure that the parameter will be valid for all unseen data to come. In fact, there might be no single parameter that suits all local patterns equally well.

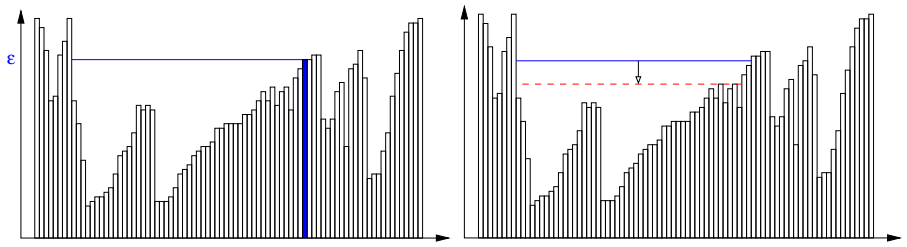
In the recent past, the *multiscale* approach has turned out to be a powerful weapon against this problem: Rather than choosing one parameter setting, examine the results for *all possible settings* and choose the single or multiple values that yield the *most stable* results. Multiscale techniques have been proven extremely helpful in many areas, such as image and shape recognition [15], signal analysis [13, 16], data compression [17], and also clustering [2, 1], to mention only a few. The next section briefly summarizes the OPTICS algorithm [1], a multiscale clustering algorithm, which will be used in the subsequent sections for pattern detection purposes.

Multiscale Clustering

In this section we will informally introduce the OPTICS algorithm, for the full details we refer to [1]. Density based clustering algorithms usually count all data objects within a hypersphere (or hyperbox) of fixed size to obtain a density estimate. We say the neighborhood $N_\varepsilon(q) = \{x \mid \|x - q\| \leq \varepsilon\}$ of a point q in the database D is dense, if $|N_\varepsilon(q)| \geq k$. Given k and ε, a cluster C is defined as a non-empty set, which satisfies two conditions: (a) a cluster has at least one point with a dense neighborhood and (b) for each point $p \in C$ with a dense neighborhood, $N_\varepsilon(p) \subseteq C$ holds. Since the identified clusters depend on the choice of ε, we speak of ε-clusters. The DB-SCAN algorithm [6] determines all clusters (with respect to ε and k) in $O(n \log n)$ where n is the number of points.

The choice of ε is crucial in the DB-SCAN algorithm, and often it is not possible to discover all the structure in a dataset with a single choice of ε. The idea of the OPTICS algorithm is to generate all partitions for all possible values of ε within some range $[0, \varepsilon_{\max}]$ (in an efficient way). But then it remains still unclear how to interpret or analyze that many resulting partitions. An interesting question to ask is at what distance ε a point p's neighborhood will become dense (called core distance) and at what distance a point p will belong to a cluster for the first time (called reachability distance). (Apparently the reachability distance is less than or equal to the core distance, because at the core distance the point will become a cluster of its own.) The OPTICS algorithm determines these two values for all data objects and, furthermore, an ordering of data objects that allows for a reconstruction of any DB-SCAN partition (see [1]). Figure 2(a)

shows an example of the so-called reachability plot, which aligns the data objects according to the determined ordering on the horizontal axis. For any point p in the plot (e.g. the marked one in Fig. 2(a)), the data points with smaller reachability values to the left make up a (DB-SCAN-) cluster at the chosen value of ε.



(a) Identification of an ε-cluster.

(b) Decreasing ε to ε' leads to embedded data subsets $I' \subseteq I$.

Fig. 2. The reachability plot (result of the OPTICS algorithm). Horizontal axis: point ordering, vertical axis: reachability value

Now it should be clear, how clusters (and local pattern candidates) are found in the reachability plot: Clusters are "dents" (or valleys) in the graph, indicating a region of high data density surrounded by data with lower density. Since the width of a valley is determined by the number of data objects in the cluster, we can use the width to distinguish large from small clusters (patterns).

5 An Approach to Local Pattern Detection

In [1] a heuristic procedure is proposed to extract clusters automatically from the reachability plot. Thresholds on the steepness and length of the flanks surrounding a flat valley are used to identify clusters. Although this technique seems to work well, a drawback is the need for selecting a new heuristic parameter.

Here, we choose a different approach. Two things are needed in order to detect substantive patterns: the pattern itself and the background. For the moment we are not concerned about *what* model we will actually use, but about the data subset that will be used to estimate the model's parameters (pattern as well as background model). A reasonable way to identify subsets is to consider all data objects that are density connected for some ε (that is, belong to the same ε-cluster). Local regions of high data density can be obtained from the reachability plot by drawing a horizontal line at ε_P. Each interval on the data axis, where the reachability plot drops below this line, corresponds to a data subset in which all points are density-reachable at ε_P. Let us denote the data objects associated

with such an interval by I_P and denote the number of points by n_P. When decreasing ε_P the subsets become more dense and smaller (cf. Fig. 2(b)).

Since we need two subsets, a larger one that corresponds to background and a smaller one that corresponds to the pattern, we simply draw another horizontal line at some larger $\varepsilon_B > \varepsilon_P$. For each local pattern subset I_P we obtain a background subset I_B with $I_P \subseteq I_B$. Now, if the pattern model P (estimated from data in I_P) deviates from the background model B (estimated from data in I_B) significantly, we have identified a substantive pattern.

This illustrates the intended approach to the detection of a substantive local patterns, but the thresholds ε_P and ε_B have not yet been determined. It is also not yet clear, how a statistical test to identify a deviation of a pattern from its background can be carried out.

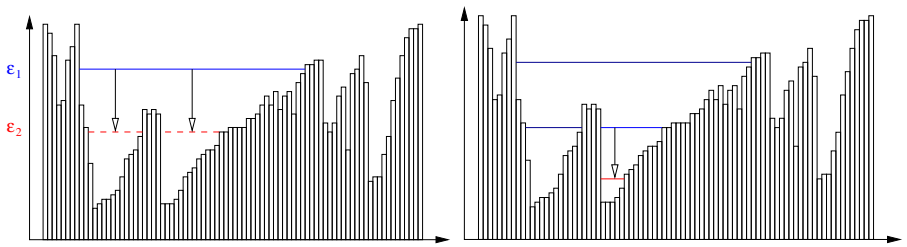
5.1 Choosing pattern and background

At the beginning, with not information available, the whole data set will be considered as the dataset for the background model (ε_B is the maximum of all reachability values). From the reachability plot we can collect all reachability values that actually occur and scan them from the largest (current background) to the smallest value. For every new value ε we pass, we have one or more data objects whose reachability value is identical to ε. Since large reachability values indicate that there is a larger gap between the data to the left and on the right, such a data point subdivides the current data subset into two or more parts (cf. Fig. 3(a)). If a statistical test (that still has to be developed) indicates that there is a significant deviation of one of the these subset from the current background, we mark this subset as a cluster (or deviation from the background). If we move the scan line further downwards, this new subset serves itself as the new background model for subsequent subdivisions, as illustrated in figure 3(b). In this way, we create a hierarchical tree of subsets directly from the reachability graph, similar to the one discussed in [13].

5.2 A Pattern Test

In the following we need local data density estimates. To calculate the data density we need to approximate the *space* that is occupied by a subset of the data. To get this estimate, we use the second outcome of the OPTICS algorithm, the core density of each data point. This is the distance to the k^{th} neighbor and can therefore be used for local data density estimation[3]. Given that for a data object x the distance to the k^{th} neighbor in the d-dimensional space is r, on average it occupies the space $V_x = \frac{V}{k}$, where $V = \frac{\sqrt{\pi^d}}{\Gamma(d/2)} r^d$ is the space occupied

[3] We used a value of 5 for k to limit the influence of border effects. Larger values are better for visual inspection of the reachability plot, but if a pattern consists of a few points only and k is high, it is very likely that the density estimation is heavily influenced by the surrounding data that do not belong to the pattern whose density we want to estimate.



(a) If ε_1 decreases, the associated data subset is split up by a peak in the reachability plot.

(b) Once a significant deviation has been identified, the newly identified pattern plays the role of the background as ε decreases further.

Fig. 3. Identification of background and pattern.

by the sphere containing the k nearest neighbors of x and Γ denotes the Gamma function. In the two-dimensional case of our illustrative examples, we assign to each data object x a volume of $V_x = \frac{\pi \cdot r^2}{k}$. The volume that is occupied by a subset of the dataset is simply the sum of volumes of each data point within the pattern or background. It should be noted that this estimation contains only the *occupied* space and free space in between is not considered. For instance, if we have two uniform clusters of identical density, the estimated volume for this data set contains the volume of the clusters only, but not the space between the clusters.[4]

There are several possibilities for defining models for patterns and background. For instance, we could use a uniform data density; we may assume that the data objects are uniformly distributed in the occupied data space, and that we have found a substantive cluster if for some subset the number of data objects differs significantly from the expected number of data objects given the volume of this subset. Having assigned data volumes V_P to the pattern and V_B to the background, we can define a binomial distribution where the probability of a randomly chosen data object lying in the pattern volume is simply $p = \frac{V_P}{V_B}$. The expected number of data objects in the pattern is then $n_B \cdot p$, which can be tested against the actual number of data objects n_P (where n_P is the number of data objects in the pattern and n_B in the background). Unfortunately, this approach fails in practice. Suppose we have a data set generated completely at random from a uniform distribution. It may happen that a few data points, say 3, are by chance very close together, much closer than the average distance between data objects. This leads to a very small total volume for this subset.

[4] This is quite different from those approaches to clustering where assumptions on certain cluster shapes are made, such as hyperspherical clusters with k-means and derivatives. There, cluster volume estimations are usually based on the center and some mean distance between data objects and center.

Any background set occupies much larger space V_B, which leads to very small pattern probabilities p. Such small probabilities make the chances of generating 3 data objects within the pattern region very unlikely even for small background sample sizes. In consequence, this approach flags much more patterns as being significant than there are actually in the dataset.

It is also possible to assume that the local data densities within a subset obey some known distribution and to test the parameters obtained from the pattern and the background for being identical. But from the construction of the subsets via the reachability plot it is clear that the pattern sample is not a random sample of the background subset, but we intentionally consider only those data values that have a small data volume. Therefore it is quite obvious that we will observe significant deviations in, say, the mean density of pattern and background quite frequently.

The approach that evaluated best is the following: Let ϱ_i be the data density estimated for data object x_i and N be the number of data objects, $\varrho_{min} = \min\{\varrho_i | 1 \leq i \leq N\}$ and $\varrho_{min} = \max\{\varrho_i | 1 \leq i \leq N\}$. The range $[\varrho_{min}, \varrho_{max}]$ of estimated data densities is partitioned into m equally sized parts

$$S_i = [\varrho_{min} + (i-1)\Delta, \varrho_{min} + i \cdot \Delta]$$

with $\Delta = |\varrho_{max} - \varrho_{min}|/m$ (in the experiments m was set to 24). We consider the local data density as being an attribute of the data object itself rather than a property of its neighborhood. Thus B (resp. P) is a m-nomial random variable whose outcome determines the density of a point in the background (resp. pattern) dataset; $P(B = S_1)$ denotes the probability of a randomly chosen data object to 'have' a data density within $[\varrho_{min}, \varrho_{min} + \Delta]$. The distribution $P(B = S_i)$ is empirically estimated from $|\{x_j | \varrho_{\min} + (i-1)\Delta \leq \varrho_j < \varrho_{\min} + i\Delta\}|/N$.

A chi-square test can be applied to test whether a sample (the pattern subset) may have been generated from this multinomial distribution. In this case, the pattern would not be flagged as a deviation from the background. But before we apply this test, we compensate for the subset selection bias mentioned in the previous paragraph. The deeper the subset is in the hierarchy (or the smaller ε_P is), the higher the data density will be. We therefore do not compare the m-nomial distributions, but exclude the part of $\mathbf{Pr}(B)$ with low data densities, which are no longer present in the subset due to the way we select the subset from the reachability graph. That is, we find a lower bound ϱ for the density values in the subset and compare $\mathbf{Pr}(B|B > \varrho)$ with $\mathbf{Pr}(P|P > \varrho)$ rather than $\mathbf{Pr}(B)$ with $\mathbf{Pr}(P)$. As an example, assume the background data density distribution is given by

$$(0.0, ..., 0.0, 0.01, 0.0, 0.03, 0.05, 0.07, 0.10, 0.09, 0.13, 0.21, 0.12, 0.11, 0.08)$$

that is $P(B = S_m) = 0.08$, $P(B = S_{m-1}) = 0.11$, etc. Starting from the left (S_0, sparse data, low data density), we calculate the number of data objects that we expect in the pattern subset with this data density, given the size $|P|$ of the current pattern candidate P. If this expected number is below 5 or no data

objects with this data density were observed in the pattern subset, the chi-square test cannot be applied and we consider a reduced $(m-1)$-nomial distribution with the leftmost slot removed. This step is repeated and the number of slots is reduced to some $0 \leq m' \leq m$. In the example, for $|P| = 100$, $m' = 9$. With $\varrho = \varrho_{min} + m' \cdot \Delta$, the distribution $P(B|B > \varrho)$ (that is, only a m'-nomial distribution) is then tested against $P(P|P > \varrho)$. This procedure is to some degree a technical necessity to apply the chi-square test, but also effectively excludes regions of low data density in the background in the comparison with the pattern candidate and thereby compensates the discussed pattern selection bias.

	number of data objects			
Figure	noise	pattern 1	pattern 2	pattern 3
5	2000	–	–	–
4(a)	2000	50	–	–
4(b)	1500	250	250	100
4(c)	1500	400	100	100
4(d)	2000	50	30	20

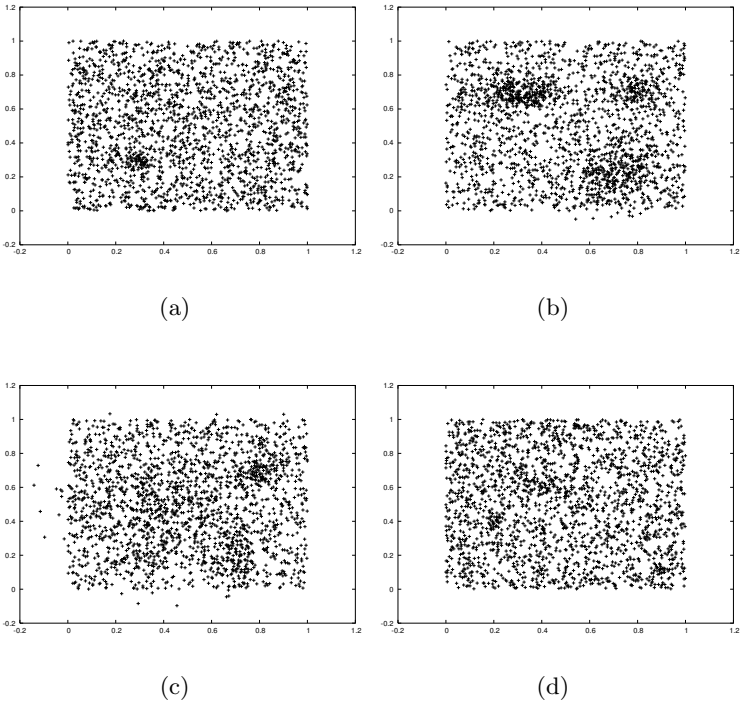
	mean values		
Figure	pattern 1	pattern 2	pattern 3
4(a)	$\binom{0.3}{0.3}$	–	–
4(b)	$\binom{0.3}{0.7}$	$\binom{0.7}{0.2}$	$\binom{0.8}{0.7}$
4(c)	$\binom{0.4}{0.5}$	$\binom{0.7}{0.2}$	$\binom{0.8}{0.7}$
4(d)	$\binom{0.4}{0.6}$	$\binom{0.2}{0.4}$	$\binom{0.9}{0.1}$

	covariances		
Figure	pattern 1	pattern 2	pattern 3
4(a)	$\begin{pmatrix} 0.025^2 & 0 \\ 0 & 0.025^2 \end{pmatrix}$	–	–
4(b)	$\begin{pmatrix} 0.1^2 & 0 \\ 0 & 0.05^2 \end{pmatrix}$	$\begin{pmatrix} 0.1^2 & 0 \\ 0 & 0.1^2 \end{pmatrix}$	$\begin{pmatrix} 0.05^2 & 0 \\ 0 & 0.05^2 \end{pmatrix}$
4(c)	$\begin{pmatrix} 0.2^2 & 0 \\ 0 & 0.2^2 \end{pmatrix}$	$\begin{pmatrix} 0.05^2 & 0 \\ 0 & 0.1^2 \end{pmatrix}$	$\begin{pmatrix} 0.05^2 & 0 \\ 0 & 0.05^2 \end{pmatrix}$
4(d)	$\begin{pmatrix} 0.05^2 & 0 \\ 0 & 0.05^2 \end{pmatrix}$	$\begin{pmatrix} 0.02^2 & 0 \\ 0 & 0.03^2 \end{pmatrix}$	$\begin{pmatrix} 0.02^2 & 0 \\ 0 & 0.02^2 \end{pmatrix}$

Table 1. Construction of the data sets in figure 4 (number of global noise points, mean and covariances of local patterns).

6 Examples

In this section we present some results obtained from the proposed local pattern detection algorithm. We discuss results for five data sets, one of them consisting of 2000 data objects uniformly distributed in the unit square. All other data sets are depicted in figure 4(a)-4(d). The dataset in Fig. 4(a) has also been used in [3]. Table 1 summarizes how the data sets have been generated. Especially Fig. 4(d) represents a difficult problem, because the superimposed patterns are really small and quite difficult to identify even for a human.

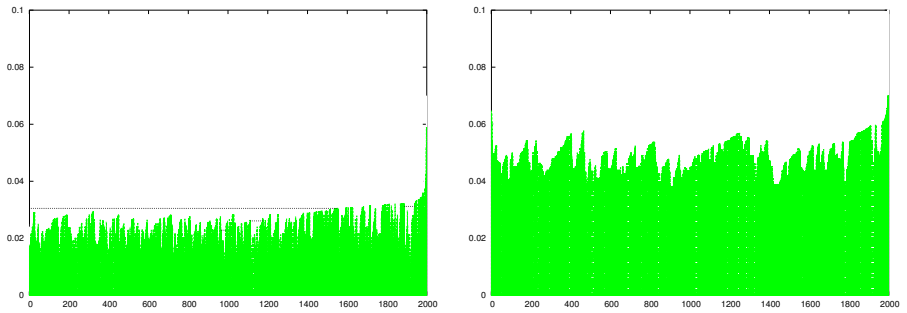


(a) (b)

(c) (d)

Fig. 4. Collection of test data sets, generated according to table 1.

Figure 5 shows the reachability graph for the uniform data set and two different values of k (number of data points in a dense neighborhood). Although no substantive patterns were superimposed over the uniform noise, the reachability plot shows many random local minima and maxima, which are more distinct for $k = 20$. For all experiments $k = 5$ has been used because 20 data points is already the size of the smallest pattern we want to discover in Fig. 4(d) (cf. also footnote 3).



(a) $k = 5$ (b) $k = 20$

Fig. 5. Reachability plot for uniform distribution with $k = 5$ and $k = 20$.

Figure 5(a) additionally shows via horizontal lines the data subsets that were identified as substantive patterns by the algorithm, as it was discussed in Fig. 3. Four such intervals have been determined, one contains almost 75% of the data set and therefore would not qualify as a small cluster or pattern. Given the number of flagged patterns reported in [3], showing only 4 substantive pattern/background-deviations (only 3 qualify as potential patterns) is an impressive small number. The four identified subsets are shown in Fig. 6. The top left figure corresponds to the long line, the top right figure to the short line to the right. The two figures in the bottom correspond to the small patterns that use the "long line" subset as the background pattern. In both of these subpatterns the data density deviates by chance significantly from the data density in the background.

For the dataset in Fig. 4(a) with a single substantive cluster, five pattern/background combinations have been identified. They are depicted in the reachability graph in figure 7. Four of the five subsets are subsets of each other (the algorithm focuses slowly on the core of the pattern), such that only the "smallest" subset qualifies as a local pattern. Two of these hierarchically embedded subsets are shown in the bottom row of images in Fig. 7, with the smallest cluster (right bottom) corresponding very well to the superimposed normal distribution. The single remaining subset is shown in the top right image, which identifies another region of particularly high data density. This pattern is not artifically generated but occurred by chance, but *only one* such incidental agglomeration has been flagged.

The results of the datasets in Fig. 4(b) and 4(c) are shown in Fig. 8 and 9, resp. In both cases we have quite large patterns, but different data densities. The data densities of the patterns in Fig. 4(b) deviate clearly from the background noise. Similar to the previous case, the algorithm determines a sequence of significant deviations that slowly focuses on a small spot, which can then be considered as a local pattern. Although the number of marked subsets is quite

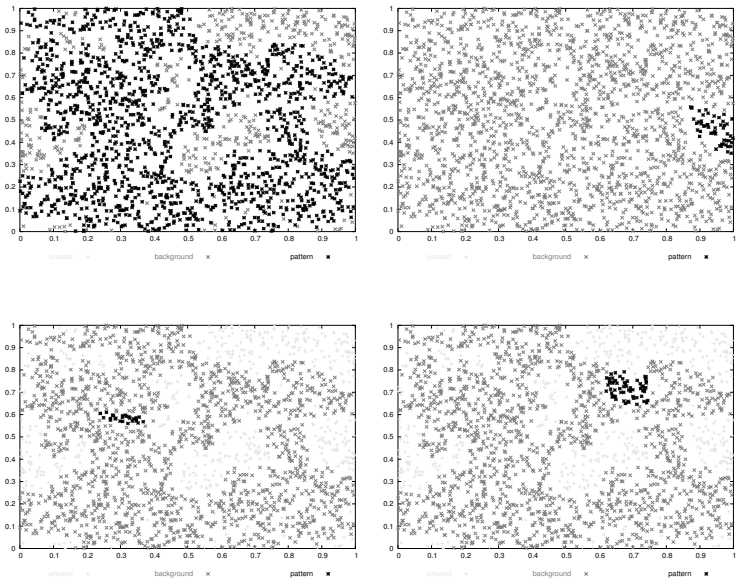


Fig. 6. Flagged clusters in the uniform data set. Top row: the whole dataset is the background data, the black points are the pattern. The pattern in the top left figure corresponds to the long line in Fig. 5(a). Bottom row: The identified pattern in the top left figure became the background (gray) for the two patterns in the bottom row. The patterns are again shown in black, the background in gray. The points in light gray do not belong to background nor pattern.

large in Fig. 8 (top left), we have only five different local patterns identified. The three largest correspond to the superimposed patterns and are shown in the figure. The fact that – compared to Fig. 8 – much more deviations have been recognized is due to the fact that Gaussian distributions have been superimposed: rather than an abrupt change in the density, which would lead to a single deviation, we have a slowly increasing data density which introduces several significant deviation levels. If we are interested in local patterns only, we can ignore all those patterns that contain an even smaller subpattern, which leads us again to a very small number of flagged local patterns.

In contrast to Fig. 4(b), the data densities of the patterns in Fig. 4(c) do not deviate that much from the background density, but this does really affect the performance of the algorithm, as we can see from Fig. 9. We have fewer focusing steps, but again the smallest patterns correspond to the superimposed Gaussian distributions. Besides the three true patterns, only one more false positive pattern has been flagged.

Finally, Fig. 10 shows the results for the most difficult test set in Fig. 4(d). Five local patterns are identified, three of them correspond to the true patterns, we have only two false positives.

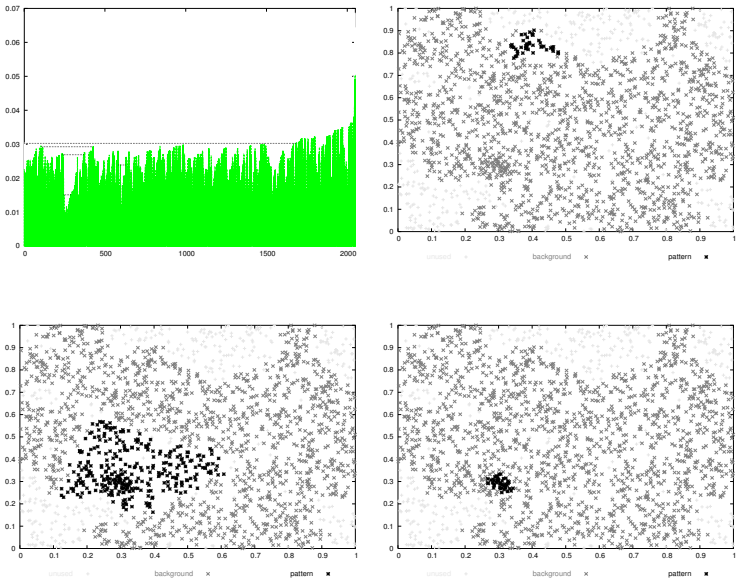


Fig. 7. Flagged clusters in the data set of Fig. 4(a). (See also explanations in Fig. 6.)

7 Summary and Conclusions

We have seen that local pattern detection (in the notion of [10]) is very similar to clustering. The challenge in local pattern detection is almost the same as in clustering, namely to identify valid, substantive structure (clusters, patterns) in data. The smaller the structure, the more difficult it is to determine its validity, because smaller structures are more likely to occur by chance.

To tackle this problem, it was proposed in [10] to install a (global) background model to verify local patterns against the background. The feasibility of the approach depends on the validity of the background model, but we have seen that we cannot restrict ourselves to simple background models. Therefore, a hierarchical approach appears to be most promising: instead of estimating a global complex background model, the utilization of a tree of simple (background) models is proposed, where each of them is installed only if it significantly deviates from the previous model. The hierarchical approach underlines that local pattern detection cannot be concerned about the local structures *only*, but has to carefully investigate structures at any scale – just like clustering.

Flagging a pattern candidate as being substantive or not is one thing, but the same care should be applied to the identification of pattern candidates (which are then passed to the statistical test). The more heuristic parameters an algorithm utilizes, the higher the chances of choosing inappropriate values. If the results are very sensitive to these parameters, how can we be sure that we identify real

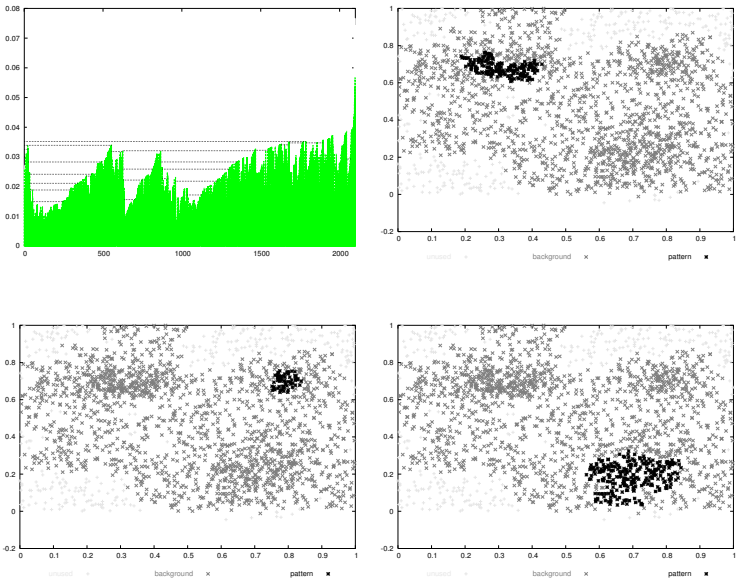


Fig. 8. Some flagged clusters in the figure 4(b). (See also explanations in Fig. 6.)

patterns or just artefacts? Multiscale algorithms have the advantage that they do not count on the user's 'guessing' capabilities but almost eliminates a threshold by analyzing the results over a large range of possible settings.

The OPTICS algorithm is a clustering algorithm that satisfies most of these requirements: it is a multiscale algorithm, is quite insensitive to the choice of the only parameter k, detects clusters of arbitrary shape and is efficient. We have discussed an alternative 'backend' to this algorithm that identifies a tree of significant deviations, whose leaves correspond to local patterns. For a number of two-dimensional test cases the results were shown: all patterns have been identified and only a very small number of false positives have been flagged. Validating the approach in a broader set of test data remains for future work.

Acknowledgments: Many thanks to Prof. Dr. Kriegel for kindly providing an implementation of the OPTICS algorithm.

References

[1] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. OPTICS: Ordering points to identify the clustering structure. pages 49–60, Philadelpha, 1999.

[2] Gerardo Beni and Xiaomin Liu. A least biased fuzzy clustering method. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(9):954–960, September 1994.

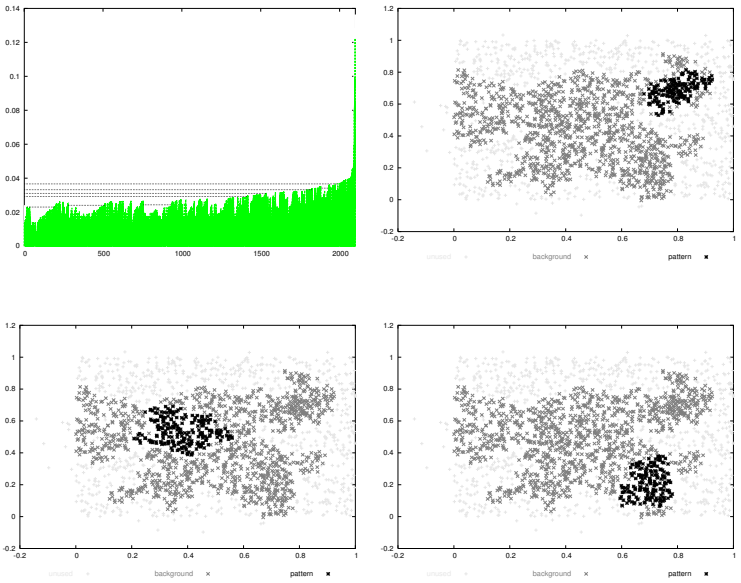


Fig. 9. Some flagged clusters in the figure 4(c). (See also explanations in Fig. 6.)

[3] Richard J. Bolton and David J. Hand. Significance tests for patterns in continuous data. In *Proc. of IEEE Int. Conf. on Data Mining*, 2001.

[4] Richard J. Bolton, David J. Hand, and Niall M. Adams. Determining hit rate in pattern search. In *[11]*, pages 36–48, 2002.

[5] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.

[6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xu Xiaowei. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–331, Portland, Oregon, 1996.

[7] B. S. Everitt. *Cluster Analysis*. John Wiley & Sons, 1974.

[8] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. MIT Press, 1996.

[9] Amir B. Geva. Non-stationary time-series prediction using fuzzy clustering. In Rajesh N. Davé and Thomas Sudkamp, editors, *Proc. of the 18th Int. Conf. of the North American Fuzzy Information Processing Society*, pages 413–417, June 1999.

[10] David Hand. Pattern detection and discovery. In *[11]*, pages 1–12, 2002.

[11] David Hand, Niall M. Adams, and Richard J. Bolton, editors. *Pattern Detection and Discovery.*, volume 2447 of *LNAI*. Springer, 2002.

[12] John A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, 1975.

[13] Frank Höppner. Handling feature ambiguity in knowledge discovery from time series. In *Proc. of 5th Int. Conf. on Discovery Science*, number 2534 in LNCS, pages 398–405, Lübeck, Germany, November 2002. Springer.

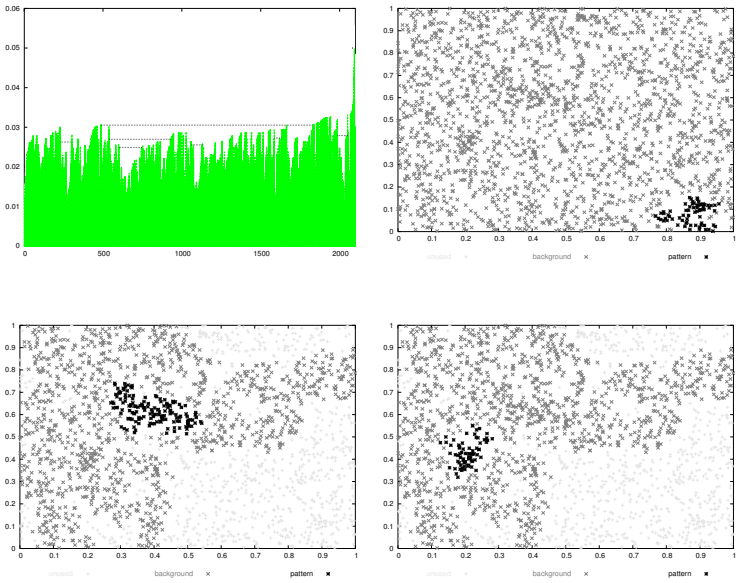


Fig. 10. Some flagged clusters in the figure 4(d). (See also explanations in Fig. 6.)

[14] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

[15] Tony Lindeberg. *Scale-Space Theory in Computer Vision*. Int. Series in Engineering and Computer Science, Robotics: Vision, Manipulation and Sensors. Kluwer Academic Publishers, Dordrecht, 1994.

[16] Stephane G. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, Inc., 2nd edition, 2001.

[17] Pabitra Mitra, C. A. Murthy, and Sankar K. Pal. Density-based multiscale data condensation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(6):734–747, June 2002.