Clustering of the points lying on monotonous curves as a partition into antichains

2015 J. Phys.: Conf. Ser. 633 012066

(http://iopscience.iop.org/1742-6596/633/1/012066)

4th International Conference on Mathematical Modeling in Physical Sciences (IC-MSquare2015) IOP Publishing
Journal of Physics: Conference Series **633** (2015) 012066 doi:10.1088/1742-6596/633/1/012066

Clustering of the points lying on monotonous curves as a partition into antichains

Eduard Lerner and Dmitry Voloskov

Kazan Federal University, Kremlevskaya str.18, Kazan 420018, Russia

E-mail: eduard.lerner@gmail.com, voloskovdmitriy@gmail.com

Abstract. Let us consider some set of points on the Cartesian plane. Each point is a part of one of few curves describing the dependency between abscissas and ordinates. In our case these are dependencies between the rock occurrence depth and the oil saturation described by Skelt-Harrison equation. In this work a problem of distributing these points into clusters corresponding to different curves is being investigated. Our original method based on presenting data points as elements of partial ordered sets with coordinate order is proposed. Thus to solve clustering problem one needs to find all the points which are parts of maximum length chains and to distribute them into corresponding antichains. One can propose obvious algorithm to solve the problem in quadratic time, based on Mirsky's theorem. In this work algorithm of $O(n \log n)$ complexity is proposed. The algorithm is based on the fact that Dushnik–Miller dimension of the partially ordered set is equal to 2 and can be applied to a wide class of dependencies.

The problem: Assume that P is a given set of distinct points (x_i, y_i), $i = 1, \ldots, n$, where n is sufficiently large (about a million or several millions). The main assumption is that there exists a small **unknown** set of continuous functions $f_j(x)$, $j = 1, \ldots, k$, whose graphs do not intersect in the considered domain, and each point (x_i, y_i) satisfies one of dependencies $y_i = f_j(x_i)$ for some j (the number k of these dependencies is also unknown). Denote the corresponding set of points by I_j, $j = 1, \ldots, k$. The simplest requirement to functions f_j is their increase. A weaker condition consists in the existence of a homeomorphism h of some domain D containing all points of P to a part of the plane D', under which all functions are increasing. This means that there exists a biunique continuous map h of the domain D to D' such that if $\{(x, y), (x'y')\} \in I_j$, then both coordinates of the difference $h(x', y') - h(x, y)$ have one and the same sign. The problem is to find the least value of k and to distribute as many points as possible over sets I_j. We propose an algorithm which solves this problem within the time of $O(n \log n + kn)$.

Note that the homeomorphism allows us to reduce to the case of increasing functions a wide class of functions such as the case when all functions satisfy the Lipschitz condition of the first order with one and the same constant L. It alows us to consider withour loss of generality only the case of increasing functions.

Let us describe one specific situation, when we needed to solve the stated problem. We were given a 70Mb array consisting of 12 million numbers representing oil saturation values in each cell of a 3D grid that described the underground part of an oil deposit. The data were provided as is, though we were not aware of the technique used for determining the oil saturation values. The problem consisted in describing the data in the shortest possible form (without loss of accuracy) in order to send them via Internet. We treated the oil saturation values in the given file as one of several (unknown to us) functions of the depth. Nonzero values started to

appear at the depth of a little less than 2 489 meters, so by adding 2 489 to the (negative) value of the depth we have concluded that the depth that corresponded to nonzero values of the oil saturation ranged from 0.236 to 112.387 meters. All these points are shown in Fig. 1.

According to the Skelt–Harrison formula [1], the dependence of the oil saturation value (y) on the depth (x) takes the form $y = a\exp\{-b/(x-d)^c\}$, $x > d$. This function equals 0 with $x = d$ and increases with $x > d$. After the performed transformations we have got $d = 0$, but the rest parameters of the curves remained unknown.

In Fig. 2 we see 6 curves. First we had to pick out (to cluster) points of these curves and then to parameterize the curves themselves. Using the algorithm described below, we have obtained 6 sets of points $I_1, \ldots, I_6$, which had to be parameterized then (each set had to be parameterized by its own curve). The number of points in the set I_1 (located on the lowest curve) appeared to be much less than the number of points located on the rest curves. Note that $\sum_i |I_i| < n$ (here n is the total number of points) therefore, having determined values of parameters of curves by the Levenberg–Marquardt method (see Fig. 2), we had to refer each of points that initially remained nonclustered to the cluster whose functional formula gave the ordinate closest to that of the point under consideration. This approach has allowed us to describe the initial 70Mb array of 12 million oil saturation values with the help of subscripts 0–6 (i.e. 1–6 — numbers of functional dependencies and 0 corresponds to zero values) and thus to reduce the size of the archive file to 679KB.

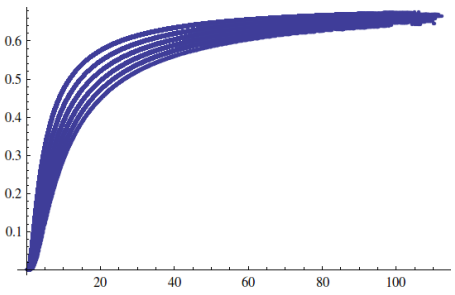

Figure 1. All points (depth, saturation) with nonzero saturation values.

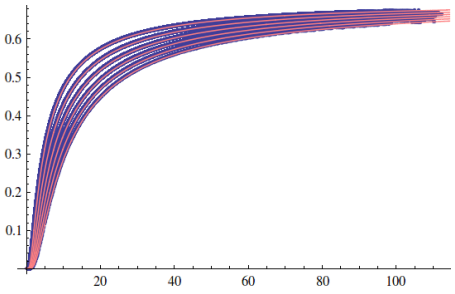

Figure 2. Approximating curves on the entire set of points (depth, oil saturation)

Before we describe the proposed method, let us consider other approaches. Note that the nearest neighbor method, which is a classical clustering technique, theoretically is applicable in our case. However, its time consumption equals $O(n^2)$, which is too much. Moreover, not everywhere in the domain under consideration the curves are separated enough.

We can simplify the problem by considering it in the characteristic part of curves (in our graph we can distinguish the domain, where the abscissa ranges from 4 to 25). The number of points in this domain is somewhat less. In view of the said above, it seems possible to solve the problem with a large number of clustered points by means of random sampling. However, according to results of our numerical experiments with the nearest neighbor method, in order to make the time consumption of the quadratic algorithm acceptable, it is necessary to reduce the sample size, at least, to several dozens of thousands of points. If the number of points on the lower curve (or the upper one) is not so much (in our specific example after the reduction of the sample size the number of such points appeared to be less than 100), then the clustering process leads to the merge of the lower curve with the upper one (because the distance between its points becomes comparable with the distance to it), or/and it falls into several parts (which has happened in our experiments).

Thus, we did not succeed in solving even the considered specific problem without several tricks. Below we describe the algorithm which solves the clustering problem for a set of points satisfying various functional dependencies in the most general case and, which is essential, does it in reasonable time.

The key moment in the formal statement of the problem, in a general case, does not consist in imposing conditions under which points can belong to one curve. On the contrary, the key moment consists in imposing conditions that make this case impossible. Note that one can numerate curves "in a natural way"; we treat the first of them as the "lower" and do the last curve as the "upper" one. Let us now introduce a partial order on the mentioned set of points; namely, let us assume that for points a and b it holds $a \succ b$, if the point a belongs to the curve that is located "above" the curve that contains the point b. Note that for "increasing" polylines a partial order is introduced as follows: for $a = (x_1, y_1)$ and $b = (x_2, y_2)$ it holds

$$a \succ b \text{ iff } x_1 < x_2,\ y_1 \geq y_2 \text{ or } x1 = x2,\ y1 > y2. \tag{1}$$

Now we can easily construct a quadratic algorithm for our problem based on the use of a POSET (a partially ordered set). Recall [2] that a chain is defined as a set of pairwise comparable elements, while an antichain is a set, all whose elements are pairwise incomparable. Thus, the problem consists in finding the least possible number k of antichains, whose union represents the whole set of points, and in finding points which (under any partition) belong to the first, second, ..., kth antichain.

Really, it is evident that the number of curves (antichains) is not less than the number of elements in the maximal chain of a POSET. Moreover, according to the Mirsky theorem [2], such a partition on an antichain exists and can be obtained by two classical methods (evidently, it is nonunique). The first technique (the ascend) implies the determination of the least elements of the POSET as points of the first antichain and their consequent deletion. Then a similar process is performed for the second antichain, and so on, until there remain only incomparable points of one kth antichain. In the second technique (the descent) we, on the contrary, start with the maximal elements of the POSET which form the kth antichain and so on.

Recall that the problem under consideration consists in finding points which under any partition belong to the first, second, ..., kth antichain. Evidently, these points exactly coincide with points which belong to some chain of length k. It is clear that points, which under both partitions belong to an antichain with one and the same number, have the mentioned property. Therefore, the intersection of two techniques for constructing a partition on an antichain mentioned in the Mirsky theorem gives a solution to our problem.

As a result, we obtain a simple algorithm. However, it is quadratic and therefore inapplicable in the case of millions of points. Note that for an arbitrary POSET we cannot propose a better variant, because it is possible that a POSET contains only two comparable elements; in this case $k = 2$ and we can refer to the first and second curve (exactly) one element, namely, each of comparable ones. Evidently, in order to find a pair of comparable elements, we are forced to enumerate all possible pairs, so the time consumption of the program is quadratic. However, in our concrete POSET there exists a certain correlation between comparable elements, and the number of them is rather large.

Let us try to describe this correlation. Assume that along with the partial order there exists a strict order $>$ agreed with the partial order in the following sense: if $a \succ b$, then $a > b$. Evidently, this assumption is not restrictive, because one can introduce the indicated order in many ways. The next assumption imposes an essential constraint. Let $a \succ b$, then $a > b$; denote by (b, a) the set of elements c satisfying the inequality $b < c < a$. Our basic assumption is that

$$\text{for any } c \in (b, a) \text{ the element } c \text{ is comparable either with } a \text{ or with } b. \tag{2}$$

If the strict order is defined, then we can sort n points in the ascending order within the time of $O(n \log n)$. Let us now discuss a linear solution algorithm for the initial problem, where the list of points π is sorted in the strict order. More precisely, below we propose a linear ascend algorithm (the descent is performed by an analogous linear algorithm). Our algorithm divides points on an antichain within one pass through the list π; its time consumption is $O(kn)$.

Evidently, the first element in the list π belongs to the first (lower) antichain. We compare the next element a with the last point c referred to the lower antichain. If it is incomparable with c (in the sense of a partial order), then we refer it to the lower antichain; but if it exceeds c in the sense of the partial order (evidently, no other case is possible), then we immediately refer it to the second antichain, if the latter is empty, otherwise we compare it with the last element of the second antichain and so on.

Let us prove that the algorithm works correctly. It suffices to make sure that the set of points referred to the lower antichain contains no extra elements. Really, let a be the first point, for which there exists an element b such that $a \succ b$. Let us first make sure that b cannot belong to the set of elements referred to the lower antichain. In this case $c \in (b, a)$, but c (by the algorithm and by the induction hypothesis) is comparable neither with a nor with b, which contradicts the main assumption. Let us now assume that b does not belong to the set of elements referred to the lower antichain. This means that there exists b' referred to the antichain such that $b \succ b'$. But then in view of the transitivity of the partial order $a \succ b'$, which contradicts the proved assertion. Since, evidently, when referring points to the set of least elements in accordance with the algorithm, we cannot miss anyone (and, as was proved earlier, there are no extra elements), the algorithm works correctly.

It remains to verify the existence of a strict order satisfying property (2) in the case of POSET (1). We can easily make sure that the inverse lexicographical order $a = (x_1, y_1) > b = (x_2, y_2)$ iff either $x1 < x2$ or $x_1 = x_2$, $y_1 < y_2$, can serve as the desired strict order.

In conclusion, let us generalize the obtained results from the point of view of the POSET theory. Thus, let a certain nonstrict order be defined. It appears to be insufficient for solving the stated algorithmic problem (to determine all elements of the POSET that belong to all maximal chains and to divide them into the corresponding antichains within a linear time or close to it). Therefore we assume that it is possible to define one more construction, namely, a strict order which agrees with the partial one and, in addition, has property (2). If such an order exists, then, as was described above, after sorting elements in this strict order, we can solve the stated algorithmic problem within a linear time.

However, it remains to solve one more problem, namely, to find necessary and sufficient conditions for introducing order (2). It appears that all such POSETs are isomorphic to the POSET of points on the two-dimensional plane with the partial order generated by the coordinatewise comparison

$$a = (x_1, y_1) \succeq b = (x_2, y_2) \text{ iff } x_1 \geq x_2, \text{ and } y1 \geq y2. \tag{3}$$

Recall that the order dimension (or the Dushnik–Miller dimension) [2] of a POSET (P, *preceq*) is defined as the least number of linear orders (chains) $(P, \preceq_i), i \in \{1, \ldots, k\}$ on P such that their intersection gives $(P, \preceq)$: $(P, \preceq) = \bigcap_{i \in \{1,\ldots,k\}} (P, \preceq_i)$ (i.e., $a \preceq b$ iff $a \preceq_i b$ for all $i \in \{1, \ldots, k\}$.

In the Dushnik–Miller paper [3] one, in fact, proves that condition (2) is equivalent to the fact that the order dimension of a POSET does not exceed two (note that in modern papers condition (2) is not used so often). It remains only to note that if the order dimension of a POSET equals m, then we can associate each element of the POSET with a point in R^m so as to make the coordinatewise order of these points coincide with the order on the corresponding elements of the initial POSET. Therefore, all POSETs with property (2) are equivalent to POSET (3).

Evidently, if the direction of the abscissa changes to the opposite one, then the partial order (1) turns into order (3). Thus, the case of increasing functions, in fact, describes the most general case, which allows the application of the developed algorithm.

The authors are grateful to Vladislav Sudakov for the provided data and to Artur Aslanyan for the problem statement.

References

[1] Skelt C and Harrison B, 1995, *Transaction of the Society of Professional Well Log Analysts* 36th Annual Logging Symposium, NNN, 10
[2] Harzheim E, 2005, Ordered Sets, *Series: Advances in Mathematics* **7** Springer-Verlag, New York
[3] Dushnik Ben and Miller E.W. 1941, *American Journal of Mathematics* **63** (3) 600