Comparison of partial orders clustering techniques

A. Raskin <a.a.raskin@gmail.com>
National research nuclear university «MEPhI»,
115409, Russia, Moscow, Kashirskoe shosse, 31.

Abstract. In this paper, we compare three approaches of clustering partial ordered subsets of a set of items. First approach was k-medoids clustering algorithm with distance function based on Levenshtein distance. The second approach was k-means algorithm with cosine distance as distance function after vectorization of partial orders. And the third one was k-medoids algorithm with Kendall's tau as a distance function. We use Adjusted Rand Index as a measure of quality of clustering and find out that clustering with all three methods get stable results when variance of number of items ranked is high. Vectorization of partial orders get best results if number of items ranked is low.

Keywords: Levenshtein distance; partial orders; clustering; distance measure: Kendall's tau distance

1. Introduction and Motivation

This investigation is a part of big project of developing clustering module for weighted sequences. As an example of such data we can suggest log of WEB site pages user opens with time, number clicks etc as characteristics of each state. Another data example (less obvious, but it is a real data we use) is set of medical treatments, provided in hospitals and polyclinics: sequence of medical treatments, which were provided to patient with a diagnosis during some fixed period of time. The main problem we try to solve is a development of system, which help specialists to analyze such sequences. One of the tools we need to implement is clustering module.

The main problem of research is a distance function between such complex-structured data. We need to take into account:

- a set of objects (e.g. medical treatments);
- parameters of objects;
- order of objects;

We start to making our own distance based on Levenshtein (it can be easily modified for our purpose), but decide to test new distance on each step to make sure, that our new distance is good enough in comparison with other distances. This paper consider first step of our research: comparison Levenshtein distance with another distances for partial orders. Partial order is simplest example of weighted sequences: there are no repeated objects and no weights.

So this paper considers the problem of clustering partial orders as a part of problem mentioned above. Since the problem of clustering orders does not differ much from the problem of clustering any set of objects we focused on distance function between objects of clustering. Comparison of partial orders obviously is quite difficult problem because if we compare two of them we need to take into account not only set of elements, but in addition an order of them. Despite complexity and interest of this theme it has surprisingly little work has been done.

We decide to compare Levenshtein distance as a function of similarity between partial orders and compare it with a recently presented approach proposed in [1] and well-known Kendall tau rank distance [5] to find out their performance in different circumstance.

2. Definitions and Problem Statement

According to [1] chain is a "totally ordered subsets of a set of items, meaning that for all items that belong to a chain we know the order, and for items not belonging to the chain the order is unknown". Hence every chain can't include one object more than one time. As an example of such data we can suggest a rating of some objects (films, music compositions etc). More precisely, when we talk about clustering chains we assume, that full data set of chains was generated from some total orders. We want to make such clusters, where all chains in one clusters were generated by one total order.

For our analysis we use Lloyd's algorithm, also known as k-means, which is one of the most common clustering algorithms and the k-medoids algorithm, which is a medoidshift clustering algorithm related to the k-means. Both the k-means and k-medoids algorithms are partional (breaking the dataset up into groups) and both attempt to minimize the distance between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the k-means algorithm, k-medoids chooses datapoints as centers (medoids or exemplars) and works with an arbitrary matrix of distances between datapoints [2]. We use two different algorithms in depend on distance function and ability to calculate mean value.

3. Distance Algorithms

As we mentioned above clustering algorithms themselves does not differ much for different objects, but the distance function highly depends on data we want to analyze. So we focused on distance function between partial orders and implement Levenshtein distance function to calculate distance between them. We also try to compare three distance functions: vectorizing algorithm presented in [1] (Ukkonen distance), Kendall's tau rank distance and our implementation of Levenshtein distance [3].

3.1. Ukkonen Distance

There were a number of different distances between partial orders in [1]. For analysis we choose planted partion model, which is very interesting first of all because it help to vectorize partial orders. It doesn't compare two orders directly, but firstly vectorize them and then use ordinary mathematical distances (Cosine, Euclidian or any other). Additionally it is very simple from computational point of view: it needs just O(nm) to compute vectors for *n* partial orders, when size of total order is m.

The main idea of planted partion model is next. A function *f* that maps total orders to R^m as follows: let τ be a total order on M, and let $\tau(u)$ denote the position of $u \in M$ in τ. Consider the vector f_τ where

$$f_\tau(u) = -\frac{m+1}{2} + \tau(u)$$

If partial orders are shorter than total order we need to take into account cases, when element from total order not exist in partial order (is not ranked). So if π is a partial order and u - one of the elements of M:

$$f_\pi(u) = \begin{cases} -\frac{|\pi+1|}{2} + \pi(u) & \text{iff } u \in \pi \\ 0 & \text{iff } u \notin \pi \end{cases}$$

And after normalization of function we get:

$$f(\pi) = {}^{f_\pi}\!/_{\|f_\pi\|}$$

After this vectorization procedure we can use any of classical distances between objects, for example, cosine distance which we use in this work. Using this distance we can use k-means algorithm, because we can easily calculate mean value of number of partial orders.

3.2 Levenshtein Distance

In information theory and computer science, the Levenshtein distance is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (insertion, deletion, substitution) required to change one word into the other.

If we think about total orders as an alphabet, partial orders as a words and elements of order as a letter we can draw full analogy from distance between partial orders to distance between words:

$$\mathrm{Lev}_{\pi,\pi'}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} Lev_{\pi,\pi'}(i, j-1) + 1 \\ Lev_{\pi,\pi'}(i-1, j) + 1 \\ Lev_{\pi,\pi'}(i-1, j-1) + 2[\pi(i) \neq \pi'(j)] \end{cases} & , else \end{cases}$$

In this case we cannot use k-means algorithm, because mean value of partial orders is not defined, so we need to use k-medoids clustering algorithm.

3.3 Kendall's Tau Rank Distance

The Kendall tau rank distance is a metric that counts the number of pairwise disagreements between two ranking lists. The larger the distance, the more dissimilar the two lists are. The main problem is that if the chains π_1 and π_2 have no items in common, we have to use a fixed distance between π_1 and π_2. For example it was made for Spearmen's rho by [4]. We can use the same approach also with the Kendall distance by defining the distance between the chains π_1 and π_2 as the (normalized) Kendall distance between the permutations that are induced by the common items in π_1 and π_2. If there are no common items we set the distance to 0.5.

The Kendall tau ranking distance between two lists L_1 and L_2 is

$$K(\tau_1,\tau_2) = \left|\{(i,j): i<j, (\tau_1(i) < \tau_1(j) \wedge \tau_2(i) > \tau_2(j)) \vee (\tau_1(i) > \tau_1(j) \wedge \tau_2(i) < \tau_2(j))\}\right|$$

where τ_1 and τ_2 are the rankings of the elements in L_1 and L_2.

4. Experiments and Results

For testing these distance functions we produce a number of clusterizations and evaluate results of clustering. We assume that quality of clusters is strongly correlated to quality of distance functions. Data we use for clustering was artificial: we generate a number of partial orders from three total orders. So we have an opportunity to use Adjusted Rand Index as a measure of quality of clustering [6,7]. For testing we make Python program in which implement K-means clustering algorithm with Ukkonen distance function, K-medoids algorithm with Kendall's tau distance and K-medoids algorithm with Levenshtein distance.

First thing we want to test is how the quality of clustering depends on fraction of items ranked. It was predictable that the bigger fraction is the easier it is to distinguish them from each other, so we produce a number of test with different fraction of items ranked. We assume that all partial orders are the same length. Results of multiple clustering tests with different number of items ranked and different number of items in total order are in fig.1

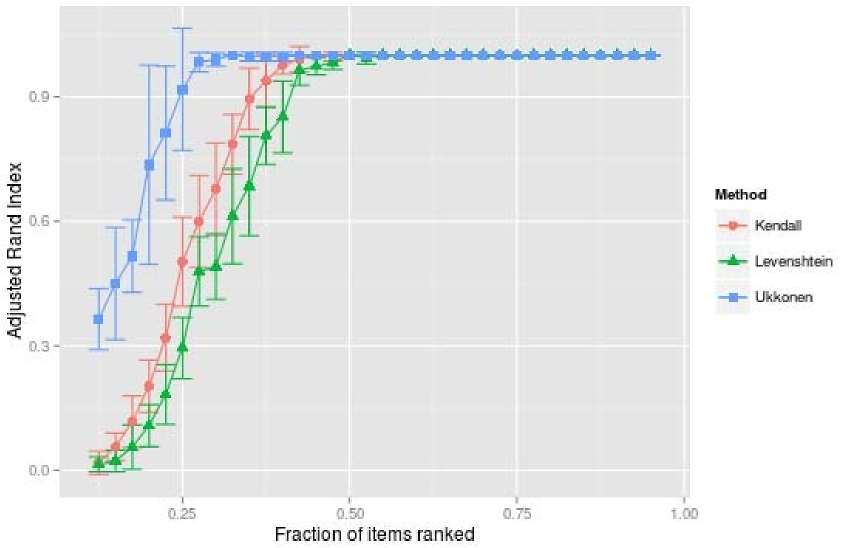

Fig. 1. Quality of clustering in depending on fraction of items ranked (all partial orders has the same length)

We can see, that if number of items ranked is equal to number of elements in total order (in other words, all elements of total order are in partial order) all three algorithms are quite good, but when partial orders are very little all of them cannot perform well.

In previous test we assume that all partial orders are of equal length. Next test helps us to define quality of distance functions in case of comparison of partial orders with different length. We want to understand if distance function can correctly compare partial orders with different number of elements. So the idea of experiment was the next one. We assume that length of chain is a random value generated by normal distribution with some mean value and some variance. The mean value is not so important in this test, because the main idea is to understand dependency of clustering quality on variance of partial orders length, so it was fixed for all experiments. Accordingly to this assumption we generate partial orders with different lengths (from normal distributions with same mean value and different variance). For each variance we evaluate Adjusted Rand Index. Results are in fig.2.

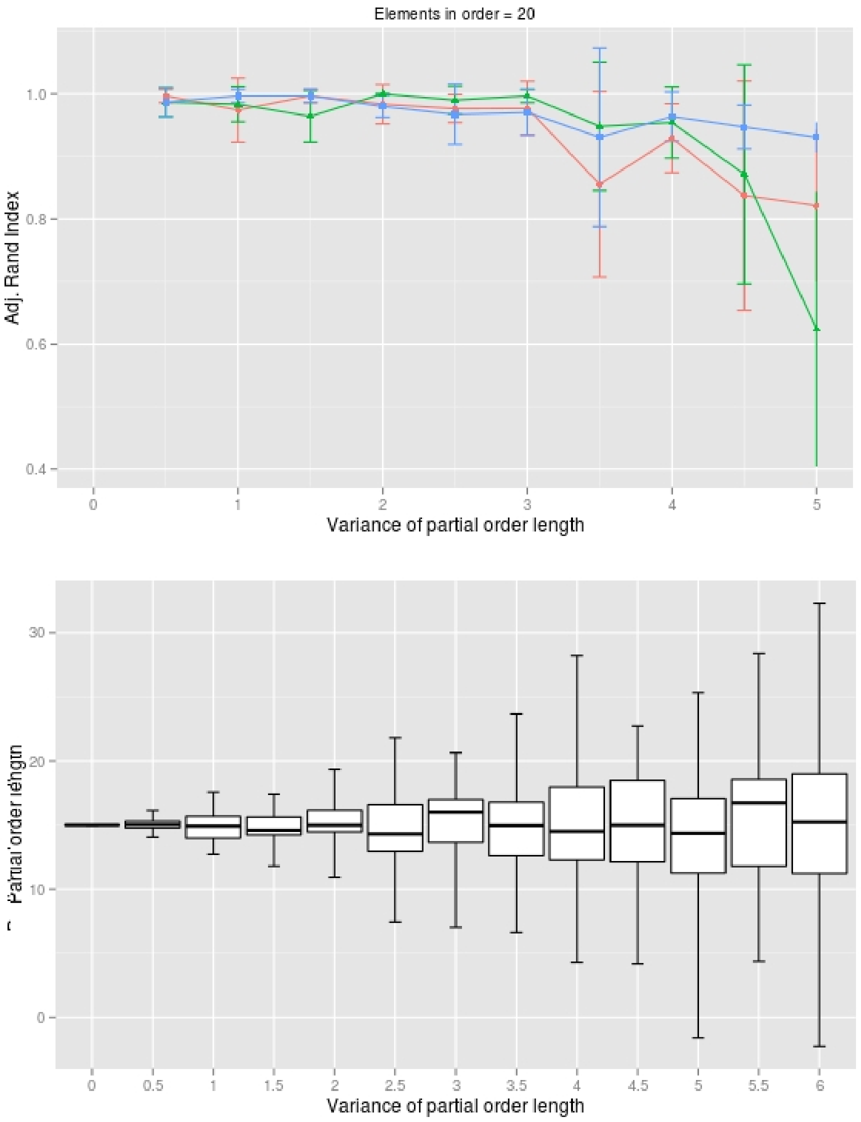

Fig. 2. Quality of clustering in depending on variance of number of items ranked.

All algorithms decreased their quality with increasing variance of number of items ranked, but we want to emphasize, that variance of clustering quality with Kendall distance significantly increase in comparison with Levenshtein and Ukkonen distances.

5. Conclusion

We find out that using Ukkonen distance help to achieve more stable results with higher quality than Levenshtein and Kendall distances. Levenshtein distance is relatively good when we take into account partial orders with the same number of elements in them. But quality of clustering process decreased with increasing variance of number of items ranked.

Kendall's tau distance get stable result with quality close to Levenshtein distance, but there is no reasonable way to modify this distance to compare weighted sequences.

We do not consider that fact in paper, but we cannot to ignore that fact that Ukkonen distance showing great promise property: we can vectorize (and in some cases vizualize) partial orders using this algorithm while Levenshtein distance is applied directly to partial orders and all problems of vizualization. Another good property is the computational complexity of the algorithm: we can vectorize n objects in O(nm), when the size of the total order is m and use after that simple functions to get distances. The main problem of such approach is necessity to know size of full order, while other distance functions has no need in such information.

References

[1]. A. Ukkonen. Clustering algorithms for chains. Journal of Machine Learning Research, 12:1389–1423, 2011.

[2]. L. Kaufman, P. Rousseeuw. Clustering by means of medoids. In Dodge, Y. (Ed.) Statistical Data Analysis based on the L1 Norm. Elsevier/North Holland, Amsterdam, 1987, pp. 405-416.

[3]. V.I. Levenshein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady. Vol. 10, No. 8, 1966, pp. 707-710

[4]. T. Kamishima, J. Fujiki. Clustering orders. Proceedings of the 6th International Conference on Discovery Science, 2003, pp.194-207

[5]. M. Kendall, J.D. Gibbons. Rank Correlation Methods. A Charles Griffin Title, 1990. 272 p.

[6]. W.M. Rand. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association. Vol. 66, No. 336, 1971, pp. 846-850

[7]. L. Hubert, A. Phipps. Comparing partitions. Journal of Classification. Vol. 2, No. 1, pp.193-218

Сравнение методик кластеризации частично упорядоченных множеств

А.А. Раскин <a.a.raskin@gmail.com>
Национальный исследовательский ядерный университет «МИФИ»,
Каширское ш., 31, Москва, 115409

Аннотация. В статье предлагается сравнение трех подходов к кластеризации частично упорядоченных множеств. Первый подход заключается в применение алгоритма кластеризации k-medoids с использованием расстояния Левенштейна. В качестве второго подхода рассматривается векторизация частично упорядоченных множеств с дальнейшей кластеризацией с помощью алгоритма k-means и косинусного расстояния в качестве функции расстояния между объектами. Последним рассматриваемым подходом является кластеризация с помощью алгоритма k-medoids и коэффициента ранговой корреляции Кендалла в качестве функции расстояния. Для оценки качества кластеризации мы использовали Adjusted Rand Index и определили, что кластеризация с использованием всех трех подходов дает стабильный результат даже в тех случаях, когда количество элементов в кластеризуемых множествах существенно различается. В случаях, когда доля ранжированных элементов мала, наилучшие результаты показывает метод векторизации частично упорядоченных множеств.

Ключевые слова: Расстояние Левенштейна; частично упорядоченные множества; кластеризация; меры близости; коэффициент корреляции Кендалла.