POIKILO: A Tool for Evaluating the Results of Diversification Models and Algorithms.

Marina Drosou[*]
Computer Science Department
University of Ioannina, Greece
mdrosou@cs.uoi.gr

Evaggelia Pitoura
Computer Science Department
University of Ioannina, Greece
pitoura@cs.uoi.gr

ABSTRACT

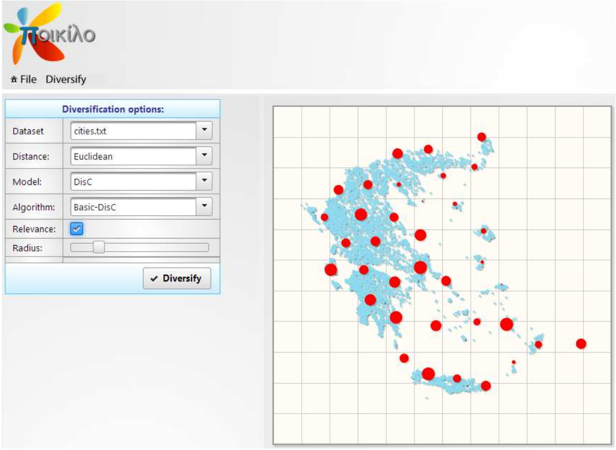
Search result diversification has attracted considerable attention as a means of improving the quality of results retrieved by user queries. In this demonstration, we present POIKILO, a tool to assist users in locating and evaluating diverse results. We provide implementations of a wide suite of models and algorithms to compute and compare diverse results. Users can tune various diversification parameters, combine diversity with relevance and also see how diverse results change over time in the case of streaming data.

1. INTRODUCTION

Result diversification has attracted considerable attention as a means of enhancing the quality of query results presented to users. Consider, for example, a user who wants to buy a new apartment in London and submits a related query to a search engine. A diverse result, i.e., a result containing various types of apartments in different London neighborhoods is intuitively more informative than a homogeneous result containing only apartments with similar features.

There have been various definitions of diversity [9], based on (i) *content* (or *similarity*), i.e., selecting items that are dissimilar to each other (e.g., [17]), (ii) *novelty*, i.e., selecting items that contain new information when compared to what was previously presented (e.g., [7]) and (iii) *semantic coverage*, i.e., selecting items that belong to different categories or topics (e.g., [4]). Most approaches rely on assigning a diversity score to each item and then selecting either the k items with the highest score for a given k (e.g., [5, 13, 14]) or the items with score larger than some threshold (e.g., [16]). Alternatively, in [10], a tuning parameter called *radius* explicitly expresses the desired degree of diversification which determines the size of the diverse set.

Different diversification methods aim at optimizing different diversification criteria. Often, it is not clear what method is more suitable for a specific application. In this demonstration, we present POIKILO (from the greek $\pi o\iota\kappa\acute{\iota}\lambda o$, meaning "diverse"), a tool to assist users in locating, visualizing and comparing diverse results based on a suite of different diversification models and algorithms. We provide implementations of a wide variety of diversification approaches for retrieving diverse results. For the case in which the degree of diversification is specified by a radius, we also provide an interactive zoom-in and zoom-out form of functionality (Figure 2).



Dataset cities.txt
Distance: Euclidean
Model: DisC
Algorithm: Basic-DisC
Relevance: ☑
Radius:

Figure 1: Poikilo user interface. Users can upload data, configure diversification options and see diversified results. Diverse results are shown as solid circles, while their size varies based on their relevance.

Often, results are associated with a relevance score. POIKILO includes various methods for combining relevance and diversity in selecting representative results. Furthermore, we consider the case of streaming data, where the query results change over time and so does the diverse result presented to the users. We employ a sliding window streaming model and provide options to navigate between consequent windows of diverse results.

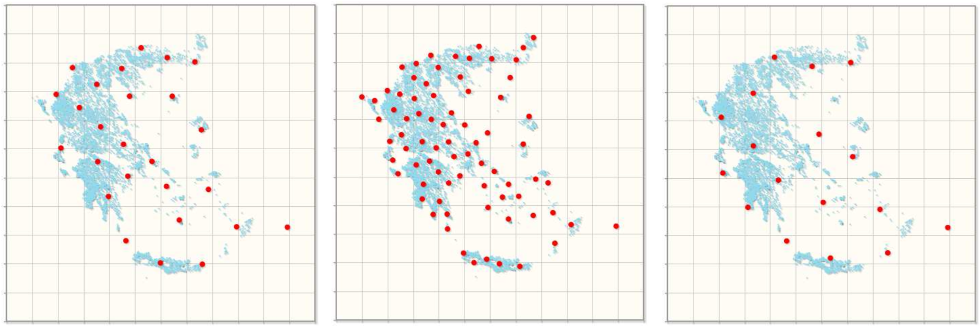
In the demonstration of POIKILO, users will be given the opportunity to submit queries to a number of different datasets and see a visualization of a diversified subset of their query result (Figure 1). We provide various synthetic and real datasets. Users can also upload their own datasets. Users can choose among a wide selection of diversification algorithms and specify various configuration parameters. Furthermore, they can zoom-in and zoom-out of this initial diverse subset and navigate between consequent windows in the case of streaming data.

2. DIVERSITY MODELS

Various models have been proposed for result diversification [9]. In this section, we describe the various models made available to users by POIKILO. Most of these models involve the use of a distance function. We have implemented the most common distance functions (e.g., Euclidean, cosine). In ad-

[*]The work of the first author was co-financed by the European Union (ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the NSRF-Research Funding Program: Heracleitus II.

Proceedings of the VLDB Endowment, Vol. 6, No. 12
Copyright 2013 VLDB Endowment 2150-8097/13/10... $ 10.00.



(a) Initial diverse items. (b) Zooming-in. (c) Zooming-out.

Figure 2: Zooming operations in action in Poikilo. Selected items are shown as solid circles.

dition, users can select which of the attributes of each item will be used for diversification.

Dispersion models. The most widespread diversity models are related to the k-dispersion problem, defined as selecting k out of a set $\mathcal{P}$ of items in some space, such that some objective function is maximized. Common variations include the MaxMin and MaxSum methods. Given a distance metric d and an integer k, $k > 1$, MaxMin aims at locating a subset S of $\mathcal{P}$ with k items, such that, the minimum pairwise distance among any items in S is maximized. MaxSum, on the other hand, aims at maximizing the sum of the respective pairwise distances. That is, the two models aim at maximizing the following diversity functions:

$$f_{\textsc{Min}}(S,d) = \min_{p_i,p_j \in S} d(p_i,p_j) \text{ and } f_{\textsc{Sum}}(S,d) = \sum_{p_i,p_j \in S} d(p_i,p_j)$$

Intuitively, MaxMin aims at discouraging the selection of nearby items, while MaxSum at increasing the average pairwise distance among all items.

DisC diversity. DisC is a recently proposed model that combines coverage and diversity [10]. Let $N_r(p_i)$ be the *neighborhood* of an item $p_i \in \mathcal{P}$, i.e., the items lying at distance at most r from p_i. r, $r \geq 0$, is a tuning parameter called *radius*. Let also $N_r^+(p_i)$ be the set $N_r(p_i) \cup \{p_i\}$. Intuitively, we would like to select exactly one item from each item's neighborhood.

Definition 1. (r-DisC Diverse Subset) *Let $\mathcal{P}$ be a set of items and r, $r \geq 0$, a real number. A subset $S \subseteq \mathcal{P}$ is an* r-Dissimilar and Covering *subset, or* r-DisC *diverse subset, of $\mathcal{P}$, if the following two conditions hold: (i) (coverage condition) $\forall p_i \in \mathcal{P}$, $\exists p_j \in N_r^+(p_i)$, such that $p_j \in S$ and (ii) (dissimilarity condition) $\forall$ p_i, $p_j \in S$ with $i \neq j$, it holds that $d(p_i,p_j) > r$.*

Given $\mathcal{P}$, we would like to select the smallest number of dissimilar and covering items.

Definition 2. (Minimum r-DisC Diverse Subset) *Given a set $\mathcal{P}$ of items and a radius r, find an r-DisC diverse subset S^* of $\mathcal{P}$, such that, for every r-DisC diverse subset S of $\mathcal{P}$, it holds that $|S^*| \leq |S|$.*

The DisC model allows an interactive mode of operation where, after being presented with an initial set of results for some radius r, a user can see either more or less results by decreasing or increasing r. Specifically, given a set of items $\mathcal{P}$ and an r-DisC diverse subset S^r of $\mathcal{P}$, we want to compute an r'-DisC diverse subset $S^{r'}$ of $\mathcal{P}$. Zooming can be global, in the sense that the radius r is modified similarly for all items in $\mathcal{P}$, or local, i.e., modifying the radius only for a specific area of the data set. To support an incremental mode of operation, the set $S^{r'}$ should be as close as possible to the already seen result S^r. Ideally, $S^{r'} \supseteq S^r$, for $r' < r$ and $S^{r'} \subseteq S^r$, for $r' > r$. Although in general there is no monotonic property among the optimal r-DisC diverse and r'-DisC diverse subsets of a set of items $\mathcal{P}$, for $r \neq r'$, we provide heuristics that achieve these requirements.

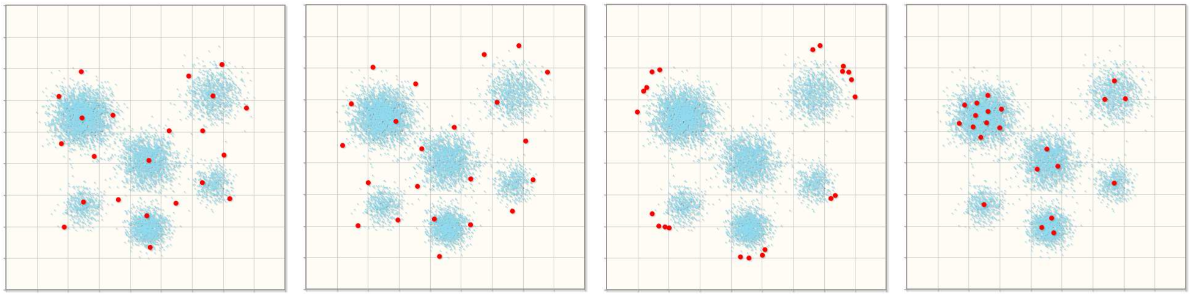
Other models. Often, clustering methods have been proposed as an alternative to selecting diverse items. In this case, the diverse set consists of representatives from each cluster. For example, k-medoids seeks to minimize $\frac{1}{|\mathcal{P}|}\sum_{p_i \in \mathcal{P}} d(p_i, c(p_i))$, where $c(p_i)$ is the closest item of p_i in the selected subset. We also consider other diversification models, such as the Greedy Marginal Contribution and Greedy Randomized with Neighborhood Expansion models presented in [15]. Our tool can be easily extended with additional methods as well.

Figure 3 shows the diverse sets located by Poikilo for some of the different approaches. Generally, MaxSum and k-medoids fail to cover all areas of the dataset; MaxSum tends to focus on the outskirts of the dataset, whereas k-medoids clustering reports only central points, ignoring sparser areas. MaxMin performs better in this aspect. However, since MaxMin seeks to retrieve objects that are as far apart as possible, it fails to retrieve objects from dense areas; see, for example, the central areas of the clusters in Figure 3. DisC gives priority to such areas and, thus, such areas are better represented in the solution. Note also that MaxSum and k-medoids may select near duplicates, as opposed to DisC and MaxMin.

3. ALGORITHMS

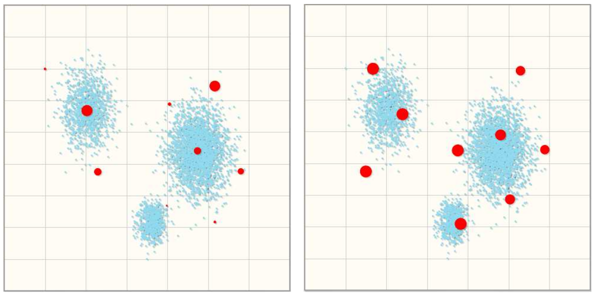
Due to the NP-hardness of most of the models of the diversification problem, a number of different heuristics have been proposed (e.g., see [12]). Poikilo provides various implementations of different variations of such heuristics.

For MaxMin and MaxSum, a simple iterative greedy heuristic has been shown to provide $1/2$-approximations of the optimal solution. In this heuristic, first, the two furthest apart items of $\mathcal{P}$ are added to S. Then, at each iteration, one more item is added to S. The item that is added is the one that has the maximum distance from the items already in S. Interchange heuristics are often used as well. Such heuristics are initialized with a random solution S and then iteratively attempt to improve that solution by interchanging an item in the solution with another item that is not in the solution. Usually, the item that is eliminated from the solution at each iteration is one of the two closest items in it. We provide various interchange heuristics, e.g., performing at each iteration the first interchange that improves the solution (*First-Interchange*) or considering all possible interchanges and perform the one that improves the solution the most (*Best-Interchange*).



(a) DisC. (b) MaxMin. (c) MaxSum. (d) k-Medoids.

Figure 3: Comparison of various diversification models.



(a) Diversity only. (b) Diversity and relevance.

Figure 4: Combining diversity and relevance. Larger item size denotes higher relevance. In (a) some areas are covered by items with very low relevance, while in (b) highly relevant items are selected.

Poikilo also provides an implementation of all the algorithms presented in [10] for computing DisC diverse subsets. These are graph-based algorithms that use a spatial index structure, namely the M-tree, to efficiently execute neighborhood queries. We briefly describe some of them next. Let us call *black* the items of $\mathcal{P}$ that are in S, *grey* the items covered by S and *white* the items that are neither black nor grey. The *Basic-DisC* heuristic initially considers that S is empty and all items are white. The algorithm proceeds in rounds; until there are no more white items, it selects an arbitrary white item p_i, colors p_i black and colors all items in $N_r(p_i)$ grey. The *Greedy-DisC* heuristic, instead of selecting white items arbitrarily at each round, selects the white item with the largest number of white neighbors, that is, the white item that covers the largest number of uncovered items. For zooming-in, i.e., for $r' < r$, we can construct r'-DisC diverse sets that are supersets of S^r by adding items to S^r. The items to be added are either selected randomly or in a greedy manner, where at each turn the item that covers the largest number of uncovered items is selected. For zooming-out, i.e., for $r' > r$, in general, there may be no subset of S^r that is r'-DisC diverse. We provide a suite of algorithms that focus on minimizing $S^r \backslash S^{r'}$, i.e., the set of items that belong to the previous diverse subset but are removed from the new one, and $S^{r'} \backslash S^r$, i.e., the set of the new items added to $S^{r'}$.

4. OTHER FEATURES

We also consider a number of aspects complimentary to diversification, namely, combining diversity with relevance and handling streaming data.

4.1 Relevance

In many cases, the items in a result set are associated with a relevance score, most often based on their relevance to the user query. In such cases, it is important to retrieve items that are highly relevant to the user query. In general, the relevance score of an item is application dependent. Without loss of generality, we assume a relevance function $w : \mathcal{P} \rightarrow \mathbb{R}^+$ that assigns a relevance score to each item, where a higher value indicates that the item is more relevant to a particular query.

Dispersion-based models combine relevance and diversity using parameters for tuning the degree of diversification. Most common approaches use weights, for example a parameter σ, $0 \leq \sigma \leq 1$, to weight the relevance of each item against its distance from other items during the selection process (a method called MMR [6]) or using a parameter λ, $\lambda \geq 0$ to favor the selection of diverse results among relevant ones. In the latter case, the corresponding relevance-aware diversity functions for MaxMin and MaxSum are:

$$f_{\text{Min}}(S,d) = \min_{p_i \in S} rel(p_i) + \lambda \min_{p_i,p_j \in S} d(p_i,p_j) \text{ and}$$

$$f_{\text{Sum}}(S,d) = (k-1) \sum_{p_i \in S} rel(p_i) + 2\lambda \sum_{p_i,p_j \in S} d(p_i,p_j)$$

In Poikilo, users can select how to combine relevance with diversity and specify the value of related tuning parameters.

Concerning the DisC model, we define the Weighted r-DisC Diverse Subset Problem:

Definition 3. (Weighted r-DisC Diverse Subset) *Given a set $\mathcal{P}$ of items with each object $p_i \in \mathcal{P}$ associated with a weight $w(p_i)$ and a radius r, find an r-DisC diverse subset S^* of $\mathcal{P}$, such that, for every r-DisC diverse subset S of $\mathcal{P}$, it holds that $\sum_{p_i \in S^*} \frac{1}{w(p_i)} \leq \sum_{p_i \in S} \frac{1}{w(p_i)}$.*

Figure 4 reports solutions for the same dataset and radius when relevance is considered or not. Again, we provide implementations of many different algorithms for handling relevance.

4.2 Streaming data

We also consider the dynamic case in which items change over time, as for example, in the case of notification services. We adopt a sliding-window model where diverse items are computed over sliding windows of length w in the input data. The length of the window w can be defined either in time units (e.g., "the most diverse items in the last hour") or in number of items (e.g., "the most diverse items among the 100 most recent ones").

We have implemented the index-based algorithms proposed in [8, 11], using Cover Trees to dynamically update the diverse subset of each window. We also provide the option to enforce the continuity properties proposed in [8, 11] among consequent windows. For example, the order in which the diverse items are delivered to the users should follow the order of their generation. Also, an item should not appear, disappear and then re-appear in the diverse set.

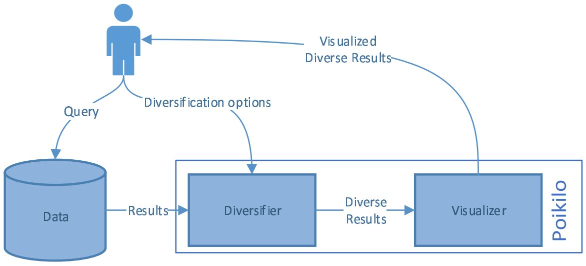


Figure 5: Poikilo system architecture.

5. DEMONSTRATION

POIKILO is a Web Application implemented in Java EE using JavaServer Faces 2.0. POIKILO can be accessed via a simple web browser using an intuitive GUI (Figure 1). The system architecture can be seen in Figure 5. During the demonstration, users will be allowed to submit queries to a number of different datasets, see diverse results and tune a variety of diversification parameters.

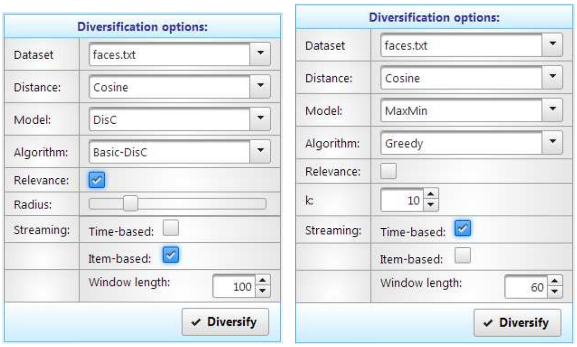
We provide a number of datasets, both real and synthetic. Our synthetic datasets consist of points in the 2D plane. Points are either uniformly distributed in space or form clusters of different sizes. Relevance scores are also assigned to items in a uniform or clustered way. We also use a number of real datasets, such as two spatial datasets containing geographic information about the location of (i) 5922 cities and villages in Greece [2], (ii) apartments in various cities (London, Paris etc.) collected from [3] and also a dataset consisting of images of people posing with different facial expressions [1]. Users can also upload their own datasets to the system via the GUI.

Upon entering the system, users are presented with a panel providing a wide variety of different diversification options (Figure 6). First, they select a dataset along with a distance metric (e.g., Euclidean, cosine, Harversine) and a diversification model (e.g., DisC, MAXMIN, MAXSUM). Then, according to the selected model, a number of algorithms and options become available to them. For example, they can select a diversification algorithm (e.g., *Basic-DisC* or *Best-Interchange*) and algorithm-specific parameters (e.g., r, k). Also, they can choose whether to also account for relevance or not during the selection of representative results and also whether to treat the input data as streaming by specifying a window length.

The computed diverse subset is presented to the users along with additional information, such as the size of the diverse subset and the average pairwise distance among the selected items. For point data, a visualization of the whole dataset is presented, in which diverse items are represented in a different size and color (Figure 1). If relevance is considered, the size of each diverse item corresponds to its relevance score, i.e., the larger this score is, the larger the item appears. Users have the option to hide the non-diverse items if they wish. For image data, the diverse set of images is presented to the user (Figure 7).

When the DisC model is employed, after being presented with the diverse subset, users have the option to tune the degree of diversification by zooming-in or zooming-out of the presented subset. A sliding bar is provided, which users can slide to dynamically increase or decrease the value of r without having to specify it explicitly.

Finally, when users use the streaming option, they have the opportunity to see how diverse items change as items enter and leave the current window by navigating between windows via “next” and “previous” buttons. Users can also request the enforcement of continuity properties among consequent windows.

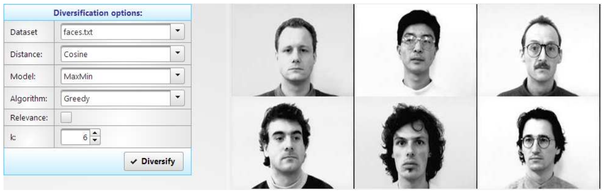

Diversification options:

Dataset	faces.txt
Distance:	Cosine
Model:	DisC
Algorithm:	Basic-DisC
Relevance:	☑
Radius:	
Streaming:	Time-based: □
	Item-based: ☑
	Window length: 100

Diversification options:

Dataset	faces.txt
Distance:	Cosine
Model:	MaxMin
Algorithm:	Greedy
Relevance:	□
k:	10
Streaming:	Time-based: ☑
	Item-based: □
	Window length: 60

Figure 6: Selecting diversification parameters.


Diversification options:

Dataset	faces.txt
Distance:	Cosine
Model:	MaxMin
Algorithm:	Greedy
Relevance:	□
k:	6

Figure 7: Diverse results for our images dataset. Users can click on an image to zoom-in and see more images similar to the one clicked.

6. REFERENCES

[1] Faces dataset. http://www.informedia.cs.cmu.edu.

[2] Greek cities dataset. http://www.rtreeportal.org.

[3] Nestoria. http://www.nestoria.co.uk.

[4] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM*, 2009.

[5] A. Angel and N. Koudas. Efficient diversity-aware search. In *SIGMOD*, 2011.

[6] J. G. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, 1998.

[7] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR*, 2008.

[8] M. Drosou and E. Pitoura. Diverse set selection over dynamic data. *IEEE Trans. Knowl. Data Eng. (to appear)*.

[9] M. Drosou and E. Pitoura. Search result diversification. *SIGMOD Record*, 39(1), 2010.

[10] M. Drosou and E. Pitoura. Disc diversity: result diversification based on dissimilarity and coverage. *PVLDB*, 6(1):13–24, 2012.

[11] M. Drosou and E. Pitoura. Dynamic diversification of continuous data. In *EDBT*, 2012.

[12] E. Erkut, Y. Ülküsal, and O. Yeniçerioglu. A comparison of *p*-dispersion heuristics. *Computers & OR*, 21(10), 1994.

[13] P. Fraternali, D. Martinenghi, and M. Tagliasacchi. Top-k bounded diversification. In *SIGMOD*, 2012.

[14] L. Qin, J. X. Yu, and L. Chang. Diversifying top-k results. *PVLDB*, 5(11):1124–1135, 2012.

[15] M. R. Vieira, H. L. Razente, M. C. N. Barioni, M. Hadjieleftheriou, D. Srivastava, C. Traina, and V. J. Tsotras. On query result diversification. In *ICDE*, 2011.

[16] C. Yu, L. V. S. Lakshmanan, and S. Amer-Yahia. It takes variety to make a world: diversification in recommender systems. In *EDBT*, 2009.

[17] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW*, 2005.