Building Representative Composite Items

Vincent Leroy, Sihem Amer-Yahia, Eric Gaussier, Hamid Mirisaee

Building Representative Composite Items

Vincent Leroy, Sihem Amer-Yahia, Eric Gaussier, Hamid Mirisaee
Université Grenoble Alpes - LIG, CNRS
Grenoble, France
Firstname.Lastname@imag.fr

ABSTRACT

The problem of summarizing a large collection of *homogeneous* items has been addressed extensively in particular in the case of geo-tagged datasets (e.g. Flickr photos and tags). In our work, we study the problem of summarizing large collections of *heterogeneous* items. For example, a user planning to spend extended periods of time in a given city would be interested in seeing a map of that city with item summaries in different geographic areas, each containing a theater, a gym, a bakery, a few restaurants and a subway station. We propose to solve that problem by building representative Composite Items (CIs).

To the best of our knowledge, this is the first work that addresses the problem of finding representative CIs for heterogeneous items. Our problem naturally arises when summarizing geo-tagged datasets but also in other datasets such as movie or music summarization. We formalize building representative CIs as an optimization problem and propose KFC, an extended fuzzy clustering algorithm to solve it. We show that KFC converges and run extensive experiments on a variety of real datasets that validate its effectiveness.

Categories and Subject Descriptors

H.3.3 [**Information storage and retrieval**]: Information Search and Retrieval—*Selection process*

Keywords

summarization; fuzzy clustering; composite items

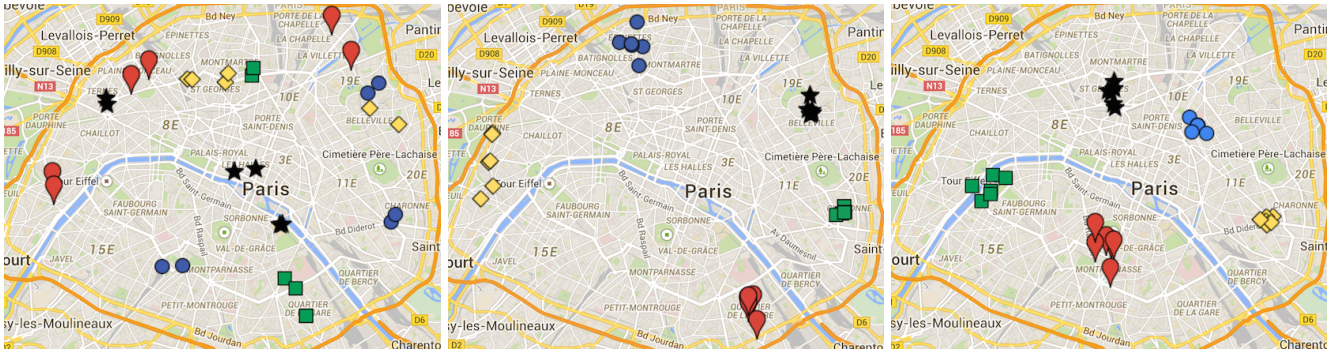
1. INTRODUCTION

The problem of summarizing a large collection of items has received a lot of attention in particular in the case of geo-tagged datasets. For example, in [13], representative tags are used to summarize a large collection of Flickr photos. In the presence of several collections, each representing a different item type (e.g., schools, restaurants, subway stations, and theaters in a city), we are faced with the question of defining an effective summarization. One could summarize one collection at a time and show the results on a map. For example, in the case of a city, each item type such as restaurant and subway station, would be summarized separately and rendered on the same map. This one-type-at-a-time approach does not necessarily guarantee that each area in the map will contain representative items of each type, nor does it ensure that items in the same area will be close to each other, i.e., *cohesive*. In this paper, we propose to explore the applicability of Composite Items (CIs) to this question.

CIs have been shown to be very effective in solving complex information needs such as planning a city tour, selecting books for a reading club, or organizing a movie rating contest [2,3,7,9,10,12,13,15,16]. In those applications, a CI is a set of close items (e.g., geographically close points of interest - POIs, movies rated by the same users) that satisfy *a budget* (e.g., at least two schools and one theater, at least one movie per genre). When summarizing POIs, a CI may correspond to geographically close places that have different types (e.g., theater and museum) and whose total visit time does not exceed 3 hours. When selecting books, a CI may be formed by similar books, i.e., on similar topics, written by different authors and whose total price is less than a maximum amount. When organizing a movie contest, a CI is a set of comparable movies, i.e., having common reviewers, and with different genres or release years. The budget constraint of a CI can therefore be used to *glue together* heterogeneous items, i.e., items with different types. In that case, we say that a CI is *valid*. The problem of summarizing heterogeneous item collections can therefore be formulated as finding K valid, cohesive and representative CIs, i.e., each CI satisfies budget constraints, is formed of "close" items, and the set of CIs "covers" all input items.

Forming valid, cohesive and representative CIs can be naturally expressed as a constrained optimization problem [2]. Existing solutions to solve this problem usually rely on two phases: in one solution, many valid CIs (i.e. satisfying the budget constraint) are built, and then the K farthest are chosen thereby resulting in representative CIs. In the other, a K-clustering is performed in the first stage to address representativity, and then one valid CI, i.e. satisfying constraints, is picked from each cluster in order to produce K CIs overall. This process decouples budget constraint satisfaction (e.g., a CI must contain one museum and 2 restaurants) from the optimization goal (e.g., each CI is a set of closely located POIs). As a result, we can argue that while clustering is a natural solution to finding CIs, existing formulations are not well-adapted to achieve validity, cohesiveness

To appear in the 24th ACM Conference on Information and Knowledge Management (CIKM), 2015, Melbourne, VIC, Australia.



(a) One-at-a-Time Approach (b) Two-Stage Approach (c) Integrated Approach

Figure 1: Alternative approaches to build composite items

and representativity simultaneously. We hence advocate the seamless integration of validity, cohesiveness and representativity when building CIs. We illustrate that on the following example.

Example 1. *Consider the case of Mary whose job is to train future users of products developed by a large software company. Mary often travels to different places where she spends extended periods of time, i.e., at least 2 weeks, during which she rents an apartment. In her free time, Mary enjoys going to the theater and dining out wherever she stays. She also practices yoga and likes swimming. Mary would be interested in exploring a map with representative CIs in different areas in the city she is planning to visit. Each CI must be valid, i.e., contain at least one theater, a pharmacy, a gym, two restaurants and a subway station (at least 6 items in total with specific cardinality constraints per item type), and cohesive, i.e., contain closeby items. Figures 1a, 1b and 1c show three sets of CIs for Paris produced using the Tourpedia dataset*[1] *with three different methods: the one-at-a-time approach that summarizes each homogeneous item collection separately, the two-stage approach that decouples validity, cohesiveness and representativity, and an integrated approach that optimizes validity, cohesiveness and representativity together. The CIs generated using the integrated approach (Figure 1c) offer the best trade-off between validity, cohesiveness and representativity. Indeed, the CIs in Figure 1a tend to favor representativity (coverage of the city) to the expense of cohesiveness (items in each CI are not close to each-other). Those in Figure 1b are located on the edges of the city because this two-stage approach first produces the most cohesive valid CIs, which limits their representativity in the second stage.*

Our example intuitively illustrates that summarizing individual lists of items and decoupling validity and representativity results in finding sub-optimal CIs in case of heterogeneous item collections. In addition, while existing approaches are based on hard clustering techniques imposing that an item must belong to one CI, in the case of a heterogeneous set of items, an item may have multiple types and should be able to belong to more than one CI. Therefore, *we propose to study the applicability of fuzzy clustering [6] to find valid and representative composite items.* In fuzzy clustering, an item may belong to more than one cluster. Our algorithm solves a joint optimization problem where one part aims at identifying item representatives (related here to cluster centroids obtained through fuzzy clustering) whereas the other part ensures that the representatives chosen are "close" to valid CIs (that is CIs satisfying budget constraints) and are cohesive. We consider in fact two problems: a minimization problem involving distance functions, and in particular the Euclidean distance, and a maximization problem involving similarity functions, and in particular cosine similarity.

Our problem naturally arises when summarizing geo-tagged datasets but also in other datasets such as book and movie summarization. That is illustrated in our experiments that make use of real datasets: **Tourpedia**[2], **BookCrossing**[3], and **MovieLens**[4]. We compare our integrated approach with the two-phase approaches proposed in [2] and show that blending validity, cohesiveness and representativity produces higher quality CIs efficiently.

In summary, the paper makes the following contributions:

- We formalize an optimization problem for building valid, cohesive and representative CIs to summarize large collections of heterogeneous items. In particular, we define validity that glues together items of different types into a single CI according to budget constraints such as:

  $$\langle 2\ drama, 2\ action, 1\ comedy, \$5 \rangle$$

  We also define a clustering objective function that captures cohesiveness and representativity. Representativity aims at finding the K CIs that cover best the input set of items. Cohesiveness is ensured by selecting the closest valid CI to each cluster centroid.

- We design `KFC`, a constraint-based fuzzy clustering algorithm that seamlessly integrates validity, cohesiveness and representativity and show that it converges.

- We run an extensive set of experiments on 3 real datasets with different characteristics. Our experiments explore the quality of CIs produced by `KFC` and compares them with state-of-the-art two-stage approaches from the literature [2]. In particular, we show that our CIs are higher quality (validity) and provide a better coverage

[1] *http://datahub.io/dataset/tourpedia*

[2] *http://datahub.io/dataset/tourpedia*

[3] *http://www.bookcrossing.com/*

[4] *https://movielens.org/*

of input items (representativity). We also run performance experiments demonstrating that KFC outperforms two-stage approaches and that it scales linearly with different parameters.

Section 2 contains our formalization and general problem statement. Section 3 describes our integrated algorithm, KFC. Experiments are provided in detail in Section 4. Related work and conclusion are given in Sections 5 and 6 respectively.

2. MODEL AND PROBLEM

In this section, we first define our formal model and discuss the link between clustering, validity, cohesiveness and representativity. We then formalize the problem of *finding a set of K, possibly overlapping, valid, cohesive and representative CIs.*

2.1 Data Model

We are given a set $\mathcal{X}$ of items where $x \in \mathcal{X}$ is uniquely identified. $\mathcal{X}$ is a heterogeneous set of items each of which may have one or several types in $\mathcal{T} = \{t_1, \ldots, t_n\}$. For example, the movie Titanic has two types: romance and drama. A book type could be novel or adventure and the type of a point of interest could be museum, park, etc. We use $x.type$ to refer to the type(s) of x. We furthermore assume that an item x may have a cost, that will be denoted as $x.cost$. For a book, this would typically be its selling price. For a museum, it could either be the cost of an entry ticket or the average time required to visit it.

We define a budget vector $b = \langle \#t_1, \ldots, \#t_n, \#_\$ \rangle$ where each $\#t_i$ specifies a cardinality for an item type $t_i \in \mathcal{T}$ and $\#_\$$ is a total cost (e.g., maximum price a user is willing to pay for a movie or maximum time a user is willing to spend visiting a place). For example, the vector $\langle 1, 2, 1, 90 \rangle$ applied on books would represent 1 novel, 2 art books, and 1 self-help book, assuming those are the only available book types, whose total price does not exceed 90$. The same vector applied on points of interest in a city would be interpreted very differently and represent 1 gym, 2 subway stops and 1 bakery and a total time not exceeding 90 minutes.

Depending on the context, we will make use of a distance function, noted $d(,)$ or a similarity function, noted $s(,)$, to compare a pair of items $(x, x') \in \mathcal{X} \times \mathcal{X}$. For instance, if x and x' are points of interest in a city, it is natural to use their geographic distance. If items are books, it is more appropriate to compare them according to content similarity, e.g. based on the cosine between their vectors; similarly, if items are movies, their similarity can be computed as the fraction of reviewers who like both x and x'.

2.2 Representativity through fuzzy clustering

We are interested in identifying valid, cohesive and representative sets of items where each item has one or several types. The *validity* of a set of items is expressed in terms of the budget vector $b = \langle \#t_1, \ldots, \#t_n, \#_\$ \rangle$ introduced above. The *cohesiveness* is the ability to identify sets of items relatively close to each other, whereas the *representativity* is the ability to cover the input dataset. The clustering literature contains many proposals for finding representative points of a dataset. Indeed, representative points are typically obtained, in any given dataset, as the centroids of the clusters present in that dataset: The set of clusters "covers" the whole dataset and their centroids represent a summary of the content of each cluster. In *hard* clustering, items are divided into distinct clusters and each item belongs to exactly one cluster, a framework well adapted to homogeneous items [13]. However, in the case of a heterogeneous set of items, an item may have different types and hence belong to more than one cluster. Therefore, we propose to study the applicability of *fuzzy* clustering [6] to the problem of finding valid, cohesive and representative items.

The most popular fuzzy clustering algorithm is Fuzzy C-Means (FCM) [6]. FCM assigns a set of items $\mathcal{X}$ to a collection of K fuzzy clusters represented through their centroids v_j, $1 \leq j \leq K$ (the set of centroids will be denoted V). More precisely, given a set of N items, $\mathcal{X}$, the algorithm returns both the K centroids and a partition matrix $W = w_{i,j} \in [0,1]$, $i \in [1, N]$, $j \in [1, K]$ where each w_{ij} represents the degree to which item x_i belongs to cluster j. Given a distance function $d(,)$, the standard objective function of FCM is as follows:

$$\underset{V,W}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{j=1}^{K} w_{ij}^m d(x_i, v_j)$$

$$s.t.\ \forall i \in [1, N],\ \sum_{j=1}^{K} w_{ij} = 1$$

where m is a weighting exponent, greater than one. A large value of m results in smaller memberships w_{ij} and hence, fuzzier clusters, whereas setting m to 1 leads to hard clustering [6]. The problem above is typically solved through an alternate optimization process in which one fixes v (respectively w) and solves for w (respectively v). The proof that such an approach converges is given in [5]; furthermore, initialization of k-means++ [4] can also be used for the centroids.

Fuzzy clustering thus represents a direct way to identify clusters in a dataset and their representative points defined by their centroids. Furthermore, its fuzzy nature enables each point to be assigned to different clusters (and centroids) through membership values.

2.3 Problem Statement

We seek to find a set of K valid, cohesive and representative items. Intuitively, validity finds sets of items that satisfy a budget constraint (i.e., cardinality and/or cost) b which glues together items of different types into *composite items*, CIs. Cohesion and representativity intuitively try to identify those CIs formed of close items that cover the input dataset (i.e., that are close to cluster centroids). We first define a valid CIs as follows:

DEFINITION 1. *Given a set of items $\mathcal{X}$ and a budget b, a valid CI, denoted $\{x_1, \cdots, x_{le}; x_i \in X, 1 \leq i \leq le\}$, is a set of items such that :*

$$\begin{cases} (i)\ \forall \#t_j \in b,\ \sum_{i=1}^{le} \mathbb{1}(t_j, x_i.type) \geq \#t_j \\ (ii)\ \sum_{i=1}^{le} x_i.cost \leq \#_\$ \end{cases}$$

where $\mathbb{1}$ is an indicator function which is 1 if both arguments are equal and 0 otherwise. le is the number of items in the CI and is such that $le \geq n$, where n is the number of type

values considered. The set of all valid CIs will be denoted as $\mathcal{V}_{CI}$.

It is easy to check whether for any b and any $\mathcal{X}$ there exists a valid CI or not (one simply gathers the $\#t_j$ cheapest items of each type t_j in b and checks their total cost). We will assume that there exists at least one valid CI for the set of items and budget constraint considered. We also assume that $\mathcal{X} \subseteq \mathbb{R}^p$. This assumption is not restrictive as most data points (or items) can be transformed so as to obtain a vector representation.

We can now formulate our problem as a joint optimization problem where one part aims at identifying good summaries (i.e. cluster centroids that are representative) of the set of items whereas the other part ensures that the representatives chosen are "close" to valid CIs, which are in turn cohesive, i.e., formed of closeby items. The closest cohesive CIs to the obtained centroids are thus valid and representative of the set of items. We in fact face two problems: a minimization problem involving the distance function $d(,)$ and a maximization problem involving the similarity function $s(,)$. Note that the weighting exponent m of the fuzzy clustering part of each problem takes values in $[1, \infty]$ for the maximization problem and in $[0, 1]$ for the minimization problem. This is simply due to the fact that the functions considered should be pseudo-convex in one case and pseudo-concave in the other. This leads to, when considering distances:

$$\begin{cases} \textbf{Distance-based formulation} \\ \underset{V,W}{\text{argmin}}(1-\lambda)\underbrace{\sum_{j=1}^{K}\sum_{i=1}^{N} w_{ij}^m d(x_i, v_j)}_{FC} + \\ \qquad\qquad \lambda \underbrace{\sum_{j=1}^{K} \min_{C \in \mathcal{V}_{CI}} \left(\sum_{x \in C} d(x, v_j) \right)}_{\texttt{CRCI}} \\ s.t.\ \forall i \in [1, N],\ \sum_{j=1}^{K} w_{ij} = 1 \end{cases} \quad (1)$$

where V denotes a set of K points (centroids) and W a partition matrix of size $N \times K$. The **Similarity-based formulation** is obtained from the above by replacing argmin by argmax, min by max, and $d(,)$ by $s(,)$. λ is a parameter that controls the influence of the two aspects of the problem: identifying cluster centroids that are representative of the complete dataset (*FC - Fuzzy Clustering*) while ensuring that the centroids obtained are close to some valid CI (`CRCI` - *Close Representative CI*). Minimizing the sum of the distances of all the items of the CI to the centroid in `CRCI` additionally ensures the cohesion of the valid CI considered. It is the compromise between these different aspects that allows one to identify valid, cohesive and representative CIs. It is important to note that the above formulation corresponds to an integrated approach that directly yields valid, cohesive and representative CIs. This contrasts with most previous solutions that rely on a two-step approach in which candidate CIs are first generated and then filtered [2].

Note that one could consider a more complex formulation with an explicit term to account for cohesion of items within CIs. Such a problem would however be more difficult to solve and would make us of an additional hyper-parameter. We rely on this study on the simpler form above in which cohesion is implicitly captured through the distance of all items to the centroid, as mentioned above.

Complexity considerations

The minimum sum of squared clustering problem (MSSC) is known to be NP-hard [1] (this problem is the one tackled by the classical k-means heuristic [14]). Setting λ to 0, m to 1 and $d(,)$ to the Euclidean distance in Problem (1) directly corresponds to MSSC (setting m to 1 transforms the fuzzy clustering defined in FC into a hard clustering problem; the fuzzy C-means algorithm becomes standard k-means in that case [6]). Hence, were we able to solve Problem (1) in polynomial time, we would be able to solve MSSC in polynomial time. Problem (1) is thus NP-hard (and so is its maximization version, i.e. its similarity-based counterpart).

Furthermore, the consideration of the cost constraint in the budget may render the minimization problem in `CRCI` NP-hard. We thus introduce below a generalization of the the above optimization problem that allows one to partly circumvent this problem.

3. ALGORITHMIC SOLUTION

We present here an algorithmic solution for the optimization problem above, focusing on the Euclidean distance for $d(,)$ and the cosine similarity for $s(,)$ as these are two widely used measures. Prior to that, we first introduce a slight generalization that partly circumvents the minimization problem in `CRCI`.

Given a set of items $\mathcal{X}$, $\mathcal{X} \subseteq \mathbb{R}^p$, a budget constraint b and the set of valid CIs $\mathcal{V}_{CI}$, let f be a function that associates to a point $v \in \mathbb{R}^p$ a valid CI from $\mathcal{V}_{CI}$: $f : \mathbb{R}^p \to \mathcal{V}_{CI}$. As before, we will denote by V a set of K points (centroids) and by W a partition (weight) matrix of size $N \times K$. We consider the following general minimization problem using the Euclidean distance:

$$\begin{cases} \textbf{Distance-based formulation} \\ \underset{V,W}{\text{argmin}}(1-\lambda)\sum_{j=1}^{K}\sum_{i=1}^{N} w_{ij}^m ||x_i - v_j||_2^2 + \\ \qquad\qquad \lambda \sum_{j=1}^{K} \sum_{x \in C_j} ||x - v_j||_2^2 \\ with:\ C_j = f(v_j) \\ and\ s.t.\ \forall i \in [1, N],\ \sum_{j=1}^{K} w_{ij} = 1 \end{cases} \quad (2)$$

As before, a **similarity-based formulation** can be obtained by replacing argmin by argmax and the Euclidean distance by the cosine similarity. In the remainder, we will use $\mathcal{G}_{eucl}(V, W, f)$ to denote $(1-\lambda)\sum_{j=1}^{K}\sum_{i=1}^{N} w_{ij}^m ||x_i - v_j||_2^2 + \lambda \sum_{j=1}^{K} \sum_{x \in C_j} ||x - v_j||_2^2$, with C_j obtained from v_j through f. $\mathcal{G}_{cos}$ is defined in the same way for the cosine similarity.

It is easy to see that Problem 2 is a generalization of Problem 1 as setting f to $f(v_j) = \min_{C \in \mathcal{V}_{CI}} \sum_{x \in C} ||x - v_j||_2^2$ in Problem 2 yields Problem 1. However, as mentioned above, this setting may not be always possible as it relies on a minimization problem that is NP-hard in the worst case. The general formulation provided in Problem 2 allows one to avoid this problem by considering general functions f

that can be computed more easily. We will come back to the choice of f in Section 3.1.

If the set V is fixed and f is given, so that C_j is known for $1 \leq j \leq K$, then $\mathcal{G}_{eucl}(V, W, f)$ is a convex function of W and the W that minimizes it can be obtained by setting the derivative of the Lagrangian of $\mathcal{G}_{eucl}$ (that integrates the constraints on W) with respect to W to 0 and solving for W. This leads to the following update rule for W (equivalent to the standard FCM update rule [6]):

$$w_{ij}^{(l+1)} = \left(\sum_{k=1}^{K} \left(\frac{||x_i - v_j^{(l)}||_2^2}{||x_i - v_k^{(l)}||_2^2} \right)^{\frac{1}{(m-1)}} \right)^{-1} \quad (3)$$

where l serves to indicate that new values are computed from known (old) ones. Similarly, for fixed W and given C_j, the function $\mathcal{G}_{eucl}(V, W, f)$ is convex in V. The values of V minimizing $\mathcal{G}_{eucl}$ are obtained by setting the derivatives of $\mathcal{G}_{eucl}$ with respect to V to 0 and solving for V, leading to:

$$v_j^{(l+1)} = \frac{(1-\lambda) \sum_{i=1}^{N} (w_{ij}^{(l)})^m x_i + \lambda \sum_{x \in C_j^{(l)}} x}{(1-\lambda) \sum_{i=1}^{N} (w_{ij}^{(l)})^m + \lambda |C_j^{(l)}|} \quad (4)$$

where $|C_j^{(l)}|$ represents the number of items in $C_j^{(l)}$.

For the valid composite item C_j associated to the centroid v_j, two cases may arise depending on the function f considered. Either the valid composite item provided by f for the new centroid $v_j^{(l+1)}$ leads to a better solution than the one associated to $v_j^{(l)}$, and it is kept, or it does not lead to a better solution, in which case the previous valid composite item is used. This can be formalized as:

$$C_j^{(l+1)} = \begin{cases} f(v_j^{(l+1)}) \text{ if } \sum\limits_{x \in f(v_j^{(l+1)})} ||x - v_j^{(l+1)}||_2^2 \\ \qquad \leq \sum\limits_{x \in C_j^{(l)}} ||x - v_j^{(l+1)}||_2^2 \\ C_j^{(l)} \text{ otherwise} \end{cases} \quad (5)$$

The above update rules guarantee that, starting with $W^{(l)}$, $V^{(l)}$ and f, one has:

$$\mathcal{G}_{eucl}(V^{(l+1)}, W^{(l+1)}, f) \leq \mathcal{G}_{eucl}(V^{(l)}, W^{(l)}, f)$$

as, for each update of W and V, the function $\mathcal{G}_{eucl}$ is minimized and does not decrease when updating the CIs provided by f. Thus, the algorithm iterating over the update rules defined by Eq. 3, 4 and 5 convergences (as $\mathcal{G}_{eucl}$ is lower bounded by 0) and provides a local minimum for the problem with the Euclidean distance.

The development for the cosine similarity is exactly the same, the convexity condition being replaced by a concavity one. We furthermore consider that all $x \in \mathcal{X}$ are normalized ($||x||_2 = 1$) and add a normalization constraint on v_j ($||v_j||_2 = 1$) so as to rely on a standard dot product ($s(x, x') = x^T x'$, where T denotes the transpose). The update rules obtained in this case are summarized below, where x_r is the r^{th} coordinate of x and m lies in the interval $[0, 1[$:

$$w_{ij}^{(l+1)} = \left(\frac{x_i^T v_j^{(l)}}{\sum_{k=1}^{K} x_i^T v_k^{(l)}} \right)^{\frac{1}{1-m}} \quad (6)$$

Algorithm 1

Input: $\mathcal{X}$, budget constraint b, K, λ, step η, procedure f
Output: Set S of K CIs

```
S ← ∅; λ' = λ; λ = 0
Initialize (e.g. through random assignment) V and W
  → V^(0), W^(0), f^(0)(V^(0)) = f(V^(0))
repeat
    repeat
        Update W through Eq. 3 (resp. Eq. 6)
        Update V through Eq. 4 (resp. Eq. 7)
        Update f(V) through Eq. 5 (resp. Eq. 8)
    until G_eucl (resp. G_cos) does not change
    λ = λ + η
until λ ≥ λ'
S ← f(V) (with the final f and V obtained)
```

$$\begin{cases} v_{jr}^{(l+1)} = \dfrac{A_{jr}}{(\sum_{r=1}^{p} A_{jr}^2)^{1/2}} \\ \text{with, for } 1 \leq r \leq p: \\ \quad A_{jr} = (1-\lambda) \sum\limits_{i=1}^{N} (w_{ij}^{(l)})^m x_{ir} + \lambda \sum\limits_{x \in C_j^{(l)}} x_r \end{cases} \quad (7)$$

$$C_j^{(l+1)} = \begin{cases} f(v_j^{(l+1)}) \text{ if } \sum\limits_{x \in f(v_j^{(l+1)})} x^T v_j^{(l+1)} \\ \qquad \geq \sum\limits_{x \in C_j^{(l)}} x^T v_j^{(l+1)} \\ C_j^{(l)} \text{ otherwise} \end{cases} \quad (8)$$

Algorithm 1 summarizes the steps followed. As one can note, we first set λ to 0 and gradually increase its value. By doing so, one first identifies fuzzy centroids that are then moved towards valid, cohesive CIs.

3.1 Choice of f

Because the budget constraints b considered here have two parts, related respectively to type cardinality and cost (see Definition 1), we rely on two scenarios associated to two different choices for f. In the first scenario, we restrict ourselves to budget constraints b that only contain type cardinality constraints: $b = \langle \#t_1, \ldots, \#t_n \rangle$. In that particular case, it is possible to efficiently compute, for any v_j, $\min\limits_{C \in \mathcal{V}_{CI}} \sum_{x \in C} ||x - v_j||_2^2$ through the following process:

1. Set $C \leftarrow \emptyset$
2. For $i = 1$ to n, add to C the $\#t_i$ items of type t_i closest to v_j
3. Return C

The function f defined by the above algorithm, the complexity of which is $\mathcal{O}(KN)$ in the worst case, directly yields the minimizer of `CRCI` in Problem 1 as there is no other valid CI closer to the given point v_j.

In the second scenario, we consider cost constraints in addition to type cardinality constraints, leading to the general budget constraint: $b = \langle \#t_1, \ldots, \#t_n, \#_\$ \rangle$. In that case one cannot directly use the above approach and we resort in this study to backtracking: we first select the closest item to a

given v_j with a type in b, and iteratively add the next closest item to v_j compatible with the constraint in b. If the cost constraint is violated, the process backtracks until all the constraints are satisfied. As mentioned in Section 2.3, for any b, the existence of a solution can be determined efficiently; the search for a valid CI is thus only performed when the existence of a solution is guaranteed. Lastly, the backtracking process may not lead to an optimal solution in the sense of the minimization problem defined in `CRCI` (Problem 1); it will nevertheless yield a valid CI (close to the centroid considered), which is required to solve Problem 2.

4. EXPERIMENTS

We report the results of an extensive experimental study on a variety of real-world datasets. We first examine the quality of our CIs and compare them to those produced by existing algorithms and then do an in-depth study of our integrated approach.

4.1 Summary of results

Our experiments confirm the superiority of the integrated approach over two-stage approaches using both problem formulations: distance minimization on POIs in a city and similarity maximization on movies. In summary, the CIs produced from POIs using the integrated approach are characterized by a better coverage of the city they belong to, and the CIs produced for movies are representative of a variety of genres and release periods. We also find that the integrated algorithm produces better values for the objective function (distance minimization or similarity maximization) than its competitors. The second half of the experiments studies scalability (in terms of response time) of the integrated algorithm for different parameter values (K and m) and total size of items and shows that it performs very well and can hence be used to build representative CIs on the fly.

4.2 Experimental setup

Our prototype is implemented using JDK 1.8.0. All scalability experiments are conducted on a 2.4 GHz Intel Core i5 with 8 GB of memory on OS X 10.9.5 operating system. Each result is an average of 10 runs with different random seeds.

Datasets.

We use three datasets with different characteristics summarized in Table 1. Section 3.1 defines two scenarios: with and without cost constraint. We apply the former to the first two dataset and the latter to the third dataset.

Tourpedia[5] contains a collection of heterogeneous POIs in various European cities gathered via Facebook and Foursquare. The latitude and longitude of each POI is available allowing us to use Euclidean distance. We collect data for Paris, Berlin, Barcelona and Amsterdam. The heat maps showing the density of POIs in each city are depicted on Figure 2. The budget vector for this use case exploits 4 POI types:

$$\langle 1\ accommodation, 2\ public\ transportation, 3\ restaurants, 2\ health\ services, \infty\rangle$$

The second dataset is **MovieLens**[6], a movie rating database. We rely on cosine similarity of user vectors to compute the similarity between movies. The budget used for **MovieLens** is defined on movie genres:

$$\langle 2\ drama, 2\ action, 1\ comedy, \infty\rangle$$

The third dataset is **BookCrossing**[7] which contains books and their user ratings, allowing cosine similarity computation. We assign a price to each book uniformly at random between 5$, 10$ and 15$ and define a budget vector on release dates:

$$\langle 3\ 1980s, 2\ 1990s, 3\ 2000s, \$70\rangle$$

Algorithms.

We denote our integrated approach presented in Section 3 as `KFC`. Our approach competes with two-stage approaches from the literature [2]: `BOBO` and `CAP`. `BOBO` is a produce-and-chose approach that first produces a large number of candidate CIs and then chooses K with the objective to maximize the distance between them. We also designed a variant of `BOBO`, called `RBOBO`, that optimizes representativity (i.e. the

[5] *http://datahub.io/dataset/tourpedia*

Dataset	# items	Obj. function
Tourpedia (POIs)	2 to 3000/city	min geo dist.
MovieLens (movies)	3,952	max review sim.
BookCrossing (books)	270,000	max review sim.

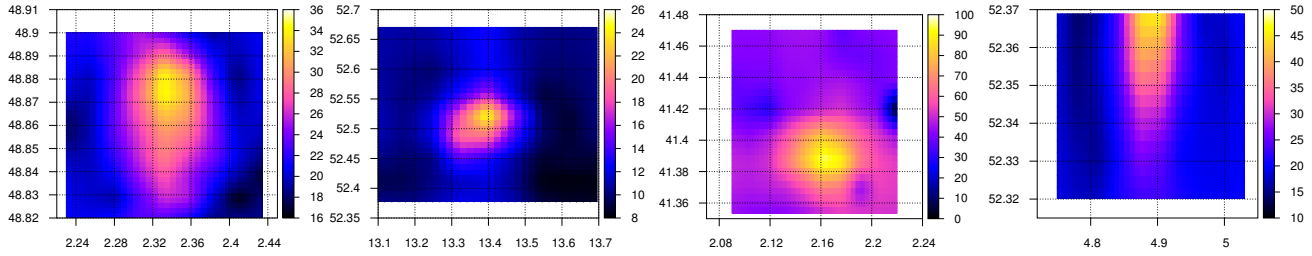
Table 1: Dataset statistics

CI	Year	Genre	Name
CI_1	1996	Drama	Big Night
	1995	Comedy	Welcome to the Dollhouse
	1991	Action	Thelma & Louise
	1997	Drama	Ice Storm, The
	1995	Action	Get Shorty
CI_2	1992	Action	Batman Returns
	1998	Drama	Truman Show, The
	1996	Drama	Jerry Maguire
	1992	Comedy	League of Their Own, A
	1994	Action	True Lies
CI_3	1990	Comedy	Back to the Future Part III
	1997	Drama	Contact
	1990	Action	Total Recall
	1995	Drama	Twelve Monkeys
	1996	Action	Independence Day (ID4)
CI_4	1998	Action	Run Lola Run
	2000	Drama	Erin Brockovich
	2000	Comedy	High Fidelity
	2000	Action	Gladiator
	1999	Drama	Magnolia
CI_5	1967	Drama	Graduate, The
	1969	Drama	Midnight Cowboy
	1973	Comedy	American Graffiti
	1971	Action	French Connection, The
	1969	Action	Butch Cassidy & Sundance Kid

Table 2: Example of results for u`KFC` with $\lambda = 0.7$ on **MovieLens**

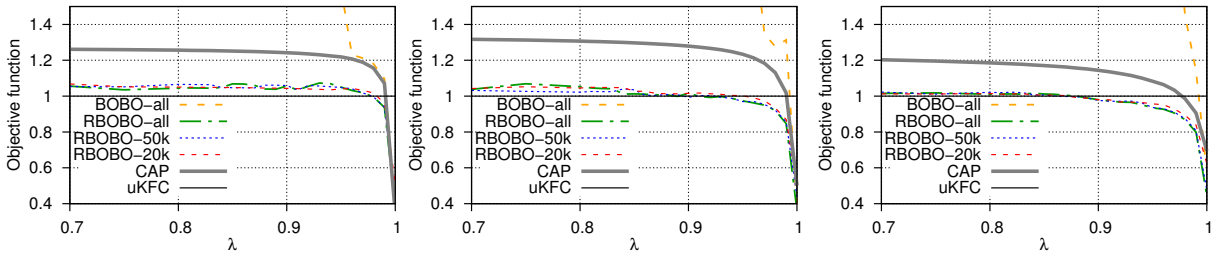
[6] *http://movielens.umn.edu*

[7] *http://www.bookcrossing.com/*



(a) Paris (b) Berlin (c) Barcelona (d) Amsterdam

Figure 2: Heat maps for the **Tourpedia** dataset with respect to latitude and longitude



(a) $K = 5$ (b) $K = 10$ (c) $K = 20$

Figure 3: Unweighted objective function with K CIs on **Tourpedia** (Paris)

distance between each item and the closest centroid of a CI) in the second phase. Iteratively, RBOBO removes the candidate that contributes the least to decreasing the distances of data points to their closest CI. Thus RBOBO relies on the same objective function as KFC. CAP is a cluster-and-pick approach that clusters data points into K sets and selects the CI whose items are most similar to each other in each set. Since these approaches do not assign weights between data points and CIs, we only compare them against uKFC, the hard clustering version of KFC (with $m = 1$). Finally, for KFC and uKFC, we set the η parameter to 0.01.

4.3 Qualitative experiment

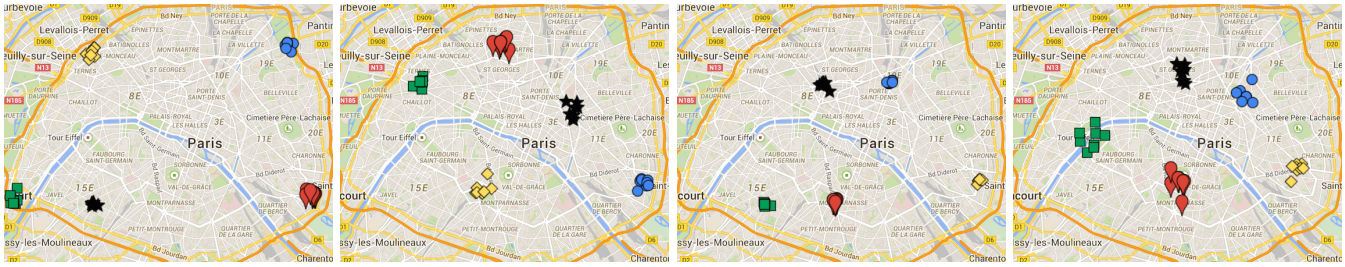
We first compare the quality of CIs produced with uKFC against those of BOBO, RBOBO, and CAP. Then, we study the quality of CIs produced by KFC in an independent quality evaluation.

4.3.1 *Comparative quality evaluation*

We use **Tourpedia** (city = Paris) to study the behavior of our objective function with different values of λ and K and for different algorithms (Figure 3). Given our problem definition, this objective function constitutes a proxy to assess the quality of the CIs generated over several executions. We then compare one instance of the CIs produced by KFC against those generated by two-stage approaches (Figure 4).

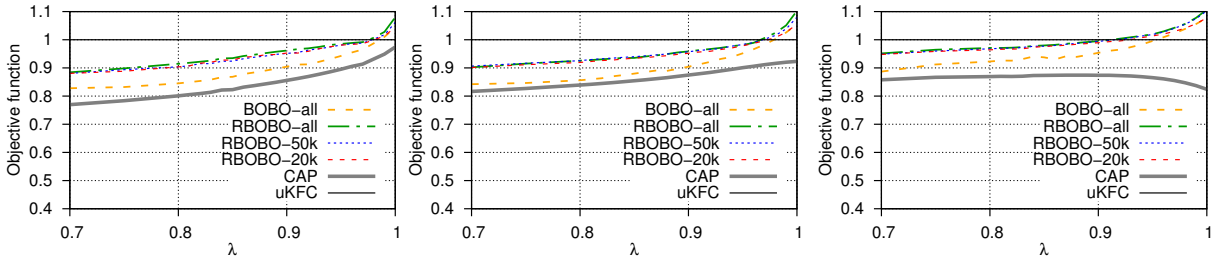
Figure 3 shows the evolution of the objective function. For better understanding of the results, we normalized each score by the one obtained by uKFC. We observe that, when λ is lower than 0.98, uKFC always produces better values of the objective function, which means that the CIs generated offer a better compromise between cohesiveness and representativity. BOBO selects K CIs by optimizing the distance between them. While it seems as it could indirectly promote representativity, in practice, BOBO tends to select outliers, and thus obtains poor results. This highlights the difference between KFC and existing approaches [2]: the objective function considers all points in the dataset, while BOBO only takes the selected CIs into account. In this evaluation, we varied the number of candidates available to RBOBO, from $20K$ candidates, to CIs formed from all data points. Our experiments show that this only marginally improves the results obtained. Since we adapted RBOBO for representativity, it significantly outperforms BOBO, but remains less efficient than KFC. CAP, while based on clustering, obtains worse results than most algorithms, as CIs selected, while originating from different clusters, can still be close to each other, which is detrimental to representativity. Overall, as K increases, the difference between uKFC and its competitor diminishes. Indeed, as more CIs are selected, each item is on average closer to a CI centroid, and thus the potential difference in the objective function is reduced. When λ nears 1, the objective function almost only takes cohesiveness. In such configurations, algorithms based on generating huge amounts of candidate CIs obtain better results, as they have a higher probability of selecting items very close to each other.

To confirm this analysis of the algorithm's behavior, we present in Figure 4 an instance of the results obtained by each algorithm on the **Tourpedia** (Paris) dataset. As explained previously, to maximize distance between CIs, BOBO selects outliers from the periphery of the city, which is clearly visible on Figure 4a. There are fewer points in the areas around those outliers (Figure 2a), so this achieves bad representativity. RBOBO does significantly better, thanks to the heuristic selecting K CIs among the candidates. The CIs selected are spread on the periphery of dense areas, so representativity is quite good, but not optimal. The results of CAP and uKFC exhibit some similarities. Indeed, they are both based on a clustering approach. However, CAP can still select CIs close to each other, which can be observed on the left of Figure 4c. On the contrary, CIs selected by uKFC are well spread over space and achieve the best representativity. We obtain similar results on the other cities.



(a) BOBO-all (b) RBOBO-all (c) CAP (d) uKFC ($m = 1$)

Figure 4: CIs selected by each algorithm with $K = 5$ CIs, $\lambda = 0.7$ on **Tourpedia** (Paris)



(a) $K = 5$ (b) $K = 10$ (c) $K = 20$

Figure 5: Unweighted objective function for K CIs with **MovieLens**

We conduct a similar experiment on the **Movielens** dataset, and present the results on Figure 5. Note that in this setup (cosine similarity), the goal is to maximize the objective function. We observe a similar trend as in the **Tourpedia** experiment. uKFC achieves a higher score as long as representativity is taken into account. When λ reaches extreme values, cohesiveness becomes the main factor, and produce-and-choose approaches obtain better results. Note that on this dataset CAP performs particularly poorly.

Due to space constraints, we are unable to present examples of the CIs composed by each algorithm. Hence, we will focus on the results of uKFC, presented on Table 2. We can see that these CIs are quite representative of different topics, styles and periods. For example, CI_3 shows mainly science fiction while CI_4 shows movies with good soundtracks.

4.3.2 *Independent quality evaluation*

We first examine the quality of the results obtained by uKFC on the **Tourpedia** Paris dataset (Figure 4d). The CIs generated are clearly localized around some of Paris' most famous attractions: the Eiffel tower, Montmartre, Montparnasse, République and Nation. Indeed, the presence of these attractions affects the distribution of POIs in Paris, and KFC is able to capture this to select representative CIs.

We now consider the results obtained by KFC for Berlin (Figure 6) and Barcelona (Figure 7) for different values of m (the weight exponent that controls the centroid positioning process). The higher m, the fuzzier representativity is. Thus, while CIs are quite spread out for low values of m, they converge towards the most central point of the dataset as m increases. This is particularly true in the case of Berlin, as the dataset contains several POIs located on the outside of the city. In practice, m could be used to control the coverage spread of the resulting CIs, while maintaining representativity.

4.4 Scalability experiment

This last set of experiments report a study of the response time scalability of KFC and RBOBO, its closest competitor, with varying values of K, m and total number of items. **BookCrossing** is the most challenging use case, as it combines a large dataset size and a total cost constraint. Figure 8a shows the execution time of KFC. Overall, we observe that KFC scales linearly with both K and m. This behavior is inherited from the clustering approach underlying KFC, and shows that while the dataset grows, convergence remains rather fast. Despite running on a laptop, KFC generates 20 composite items on a dataset of 250,000 items in less than 7 minutes.

The results for RBOBO are given on Figure 8b. RBOBO computes the distance between each candidate CI and each item. In the case of a large dataset, this can quickly become overwhelming. Furthermore, as the size of the dataset increases, the number of candidate CIs increases in order to preserve results quality. RBOBO iteratively removes candidates until reaching the desired number of K CIs. Hence, its execution time is driven by the number of candidates, and not by K. Consequently, we were unable to execute RBOBO for more than 5000 items, as the execution was lasting over 10 minutes.

5. RELATED WORK

To the best of our knowledge, this is the first study focusing on the identification of "representative" composite items of a heterogeneous set of items. While there is no work that can be directly compared to ours, there are multiple related areas. We provide a brief summary of each area.

Photo Summarization. The problem of summarizing a large collection of homogeneous items has received a lot of attention in particular, for photo summarization. Different metadata was used ranging from location to temporal

(a) $m = 2$ (b) $m = 3$ (c) $m = 4$

Figure 6: CIs generated by KFC with $\lambda = 0.7$ and $K = 5$ on **Tourpedia** (Berlin)

(a) $m = 2$ (b) $m = 3$ (c) $m = 4$

Figure 7: CIs generated by KFC with $\lambda = 0.7$ and $K = 5$ on **Tourpedia** (Barcelona)

data to sharing information. In [13], location information was used to generate photo summaries and visualize them on a map. In [12], representative photos are generated for a given time period using patterns in photo-taking habits (later studied in *http://www.hpl.hp.com/techreports/2003/HPL-2003-165.pdf*). In [11], the authors relied on event detection in personal photo collections (e.g., birthdays) which could be used for collection summarization. In [13], summaries of POIs are generated to aid visualization on a map. All cited approaches provide summaries of homogeneous item collections as opposed to our work where we are able to summarize heterogeneous item collections.

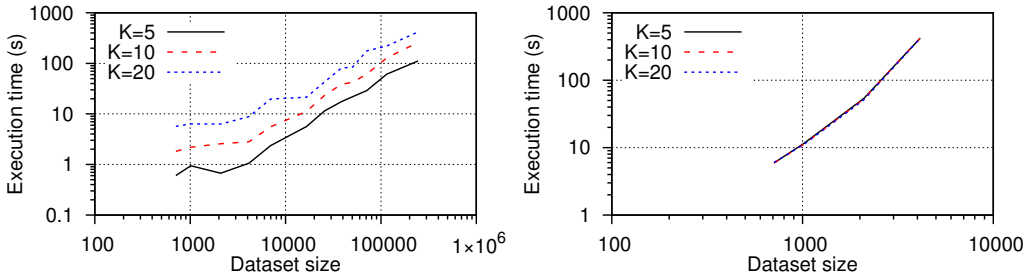
Composite Items. Composite retrieval was studied with different semantics in recent studies [2,3,7,9,10,15,16]. In [2], different algorithms were explored to build composite items. In [8], the authors propose a formalization for composite items (in the case of city tours) that is personalized. Most existing algorithms rely on a two-stage process that decouple constraint satisfaction (e.g., a CI must contain one museum and 2 restaurants) from the optimization goal (e.g., each CI is a set of closely points of interest in a city). In this paper, we show that our integrated approach is more effective than a two-stage approach. The main difference however between our approach and previous ones on composite items lies in the objective functions used for the optimization. To our knowledge, our approach is the first one to fully address the problem of identifying "representative" composite items of a set of heterogeneous items. As discussed before (Section 2), this is achieved by relying on the identification of the centroids of fuzzy clusters that are close to all the items.

Constrained Clustering. We adapt the Fuzzy C-Means (FCM) Algorithm [5] to formalize our problem. More precisely, we extend the standard Fuzzy C-Means formulation with an extra term aiming at "pushing" centroids towards valid CIs. It is that extra term that integrates the budget constraint, that guarantees the identification of valid CIs next to representative centroids, these latter ones being mainly identified as in Fuzzy C-Means. The budget constraint is integrated in this extra term via the procedure f discussed in Section 3 that associates to any given point a "close" and valid CI.

6. CONCLUSION

We explored the use of valid and cohesive composite items (CIs) to represent large collections of heterogeneous items such as POIs in a city or movies with different release dates. Validity is achieved by summarizing items with different types into a single CI. Indeed, CIs can naturally express gluing together items of different types in a budget vector such as $\langle 2\ drama, 2\ action, 1\ comedy, \$5\rangle$ where each entry specifies the minimum number of each item type desired in a CI and an upper-bound of the total cost a user is willing to pay for that CI. Cohesion and representativity are achieved by finding the best K valid CIs according to an objective function. We hence formalized the problem of summarizing large collections of heterogeneous items as that of building the K most valid, cohesive and representative CIs. Our formalization relies on two commonly-used objective functions: distance and similarity. Distance is naturally used for POIs in a city in order to find the K CIs that cover the city best. Similarity is well-adapted to representing movies in order to find the K CIs that are reviewed by similar users.

We designed a new integrated algorithm that builds the K most valid, cohesive and representative CIs of an input dataset. Our algorithm integrates constraint satisfaction into fuzzy clustering in order to simultaneously optimize for validity, cohesion and representativity. Experiments on real datasets showed that the integrated approach outperforms



(a) KFC (b) RBOBO-all

Figure 8: Scalability on **BookCrossing**

two-stage ones resulting in CIs that achieve very good representativity of existing items.

Our immediate future plans include running more extensive experiments with user studies in order to refine our notions of validity and representativity in different applications. We are also working on a formalization of our problem that admits adaptive constraints for validity thereby capturing a wide variety of interests in different item types, and on an extension in which cohesion is explicitly modeled.

Acknowledgments

This work was partially funded by ANR-13-CORD-0020.

7. REFERENCES

[1] D. Aloise, A. Deshpande, P. Hansen, and P. Popat. NP-hardness of euclidean sum-of-squares clustering. *Mach. Learn.*, 75(2):245–248, May 2009.

[2] S. Amer-Yahia, F. Bonchi, C. Castillo, E. Feuerstein, I. Méndez-Díaz, and P. Zabala. Composite retrieval of diverse and complementary bundles. *IEEE Trans. Knowl. Data Eng.*, 26(11):2662–2675, 2014.

[3] A. Angel, S. Chaudhuri, G. Das, and N. Koudas. Ranking objects based on relationships and fixed associations. In M. L. Kersten, B. Novikov, J. Teubner, V. Polutin, and S. Manegold, editors, *EDBT*, volume 360 of *ACM International Conference Proceeding Series*, pages 910–921. ACM, 2009.

[4] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[5] J. C. Bezdek. A convergence theorem for the fuzzy ISODATA clustering algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2(1):1–8, 1980.

[6] J. C. Bezdek, R. Ehrlich, and W. Full. FCM: The Fuzzy c-Means Clustering Algorithm. *Computers & Geosciences*, 10(2-3):191–203, 1984.

[7] H. Bota, K. Zhou, J. M. Jose, and M. Lalmas. Composite retrieval of heterogeneous web search. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pages 119–130, 2014.

[8] I. R. Brilhante, J. A. F. de Macêdo, F. M. Nardini, R. Perego, and C. Renso. Where shall we go today?: planning touristic tours with tripbuilder. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 757–762, 2013.

[9] A. Brodsky, S. M. Henshaw, and J. Whittle. Card: a decision-guidance framework and application for recommending composite alternatives. In P. Pu, D. G. Bridge, B. Mobasher, and F. Ricci, editors, *RecSys*, pages 171–178. ACM, 2008.

[10] M. D. Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. Automatic construction of travel itineraries using social breadcrumbs. In M. H. Chignell and E. Toms, editors, *HT*, pages 35–44. ACM, 2010.

[11] M. L. Cooper, J. Foote, A. Girgensohn, and L. Wilcox. Temporal event clustering for digital photo collections. In *Proceedings of the Eleventh ACM International Conference on Multimedia, Berkeley, CA, USA, November 2-8, 2003*, pages 364–373, 2003.

[12] A. Graham, H. Garcia-Molina, A. Paepcke, and T. Winograd. Time as essence for photo browsing through personal digital libraries. In *ACM/IEEE Joint Conference on Digital Libraries, JCDL 2002, Portland, Oregon, USA, June 14-18, 2002, Proceedings*, pages 326–335, 2002.

[13] A. Jaffe, M. Naaman, T. Tassa, and M. Davis. Generating summaries and visualization for large collections of geo-referenced photographs. In *Proceedings of the 8th ACM SIGMM International Workshop on Multimedia Information Retrieval, MIR 2006, October 26-27, 2006, Santa Barbara, California, USA*, pages 89–98, 2006.

[14] J. B. MacQueen. Some methods for the classification and analysis of multivariate observations. In *Proceedings of the 5^{th} Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[15] S. B. Roy, S. Amer-Yahia, A. Chawla, G. Das, and C. Yu. Constructing and exploring composite items. In A. K. Elmagarmid and D. Agrawal, editors, *SIGMOD Conference*, pages 843–854. ACM, 2010.

[16] M. Xie, L. V. Lakshmanan, and P. T. Wood. Breaking out of the box of recommendations: From items to packages. In *RecSys*, 2010.