Consensus Clustering Based on a New Probabilistic Rand Index with Application to Subtopic Retrieval

Claudio Carpineto, *Member*, *IEEE*, and Giovanni Romano

Abstract—We introduce a probabilistic version of the well-known Rand Index (RI) for measuring the similarity between two partitions, called Probabilistic Rand Index (PRI), in which agreements and disagreements at the object-pair level are weighted according to the probability of their occurring by chance. We then cast consensus clustering as an optimization problem of the PRI value between a target partition and a set of given partitions, experimenting with a simple and very efficient stochastic optimization algorithm. Remarkable performance gains over input partitions as well as over existing related methods are demonstrated through a range of applications, including a new use of consensus clustering to improve subtopic retrieval.

Index Terms—Consensus clustering, Rand index, probabilistic Rand index, search results clustering, subtopic retrieval

1 Introduction

Clustering is concerned with grouping objects into clusters with high intracluster similarity and low intercluster similarity. Since the definition of similarity is elusive, cluster recognition is an inherently subjective process, especially for high-dimensional data. It is this difficulty that has resulted in thousands of clustering algorithms that have been published in the last 50 years and continue to appear [1].

As a matter of fact, no clustering algorithm works well for all data. The properties of the grouping criterion used by a specific clustering algorithm may not adequately reflect either the intrinsic properties of the data or those properties that are of interest to certain users. The ill-posed nature of the clustering process, together with the success of the combination of supervised classifiers, have spurred a burst of research in recent years on consensus clustering, also known as clustering ensembles, e.g., [2], [3], [4], [5], [6], [7]. Given a set of input partitions over some objects, a consensus clustering is a partitioning of the objects which matches as closely as possible the given input partitions. By keeping the features common to several methods, consensus clustering ensures more stable and reliable decisions as well as increased robustness with respect to errors made by individual methods.

The problem of consensus clustering has been approached from various perspectives (e.g., graph-based, statistical, and combinatorial), using, among others: hypergraph partitioning [3], coassociation matrix [2], mixture model [8], and Bayesian approach [9]. The best known strategy probably consists of finding the *median* partition [18], that is, the partition that maximizes the similarity to the given partitions. Inspired by the Rand Index (RI) [13], the similarity between two partitions is usually based on the decisions made on individual pairs of objects, e.g., counting how many times the partitions agree or disagree as to whether a pair of objects should be clustered together or not.

The problem with Rand and other pair counting-based indices is that they treat all agreements or disagreements equally, even when there is a high probability that they occurred by chance. In this paper, we formalize the intuition that certain agreements or disagreements are more (less) likely and therefore they should receive less (more) credit. We introduce a new index, called the Probabilistic Rand Index (PRI), in which agreements and disagreements are weighted according to the probability of their occurring by chance. This is our first main contribution.

Our approach has been developed with a view to its applicability in a real-world scenario. Although consensus clustering techniques proved to be very useful, especially in laboratory experiments on numerical data, their deployment in real application fields raises new challenges. Among the major requirements that must be met are the following:

- Only the clustering results are accessible (not the raw data or the clustering algorithms themselves) to enable integration of results even when major proprietary, privacy, or storage issues would arise.
- The consensus clustering algorithm must be highly efficient to support real-time processing.
- The input partitions may have a different number of clusters because this parameter is specific to the particular base clustering algorithm being used.
- The number of clusters in the consensus partition should not be prespecified because it depends on the specific task and data at hand.

• *The authors are with the Fondazione Ugo Bordoni, Viale del Policlinico 147, 00161 Roma, Italy. E-mail: {carpinet, romano}@fub.it.*

Manuscript received 27 Apr. 2011; revised 28 Dec. 2011; accepted 15 Mar. 2012; published online 21 Mar. 2012.
Recommended for acceptance by F. Bach.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2011-04-0264.
Digital Object Identifier no. 10.1109/TPAMI.2012.80.

We present a full method that can deal with all these requirements. We cast consensus clustering as an optimization problem of the PRI value between a target partition and a set of given partitions. We show that a simple stochastic optimization algorithm delivers reasonable approximations of the optimal value very efficiently. The consensus clustering method is the second main contribution of this paper.

A third, major result is then represented by the experimental part. The evaluation of consensus clustering techniques typically focuses on ground truth validation, i.e., assessing how good a clustering method is at recovering known clusters from a gold standard partition. We follow the same approach, making use of artificial and text data, and, in addition, we discuss a new application of consensus clustering, namely subtopic retrieval.

The goal of subtopic retrieval is to quickly retrieve as much information as possible about the single interpretations of ambiguous or broad web queries, and this is usually achieved by means of clustering the search results retrieved by a web search engine in response to a user query [10]. The input partitions used in our experiments are acquired using recent, suitable algorithms for clustering short textual descriptions and creating meaningful cluster labels. To ensure that the clusters in the consensus partition are labeled with linguistic descriptions of high quality, we provide a method for labeling the generated clusters with the most agreed upon cluster labels.

We show that the use of the PRI-based consensus clustering method leads to an impressive average retrieval performance, with highly remarkable gains over that achieved by the input partitions as well as by existing related consensus clustering methods. Our experimental results pave the way for the realization of a *meta* clustering engine, much in the same spirit as multiple plain search engines have been combined into a meta search engine.

The remainder of the paper has the following organization: We first discuss related work, introducing background concepts and notations. Then we present the definition of the Probabilistic Rand Index and analyze its features. The next section is devoted to the PRI-based consensus clustering method, including the optimization framework, a study of approximate solutions to the optimization problem, and an illustrative example with a noisy synthetic dataset. We then turn to the experimental part. We present three comparative performance evaluation studies, two focusing on ground truth validation (with nonconvex clusters and web data) and one describing the subtopic retrieval application, and provide a discussion of the generality of the results. We finally offer some conclusions.

2 Related Work

Clustering ensemble algorithms have received a great deal of attention in the past few years. Early work has tried to solve the problem by means of clustering itself. Using a *co-association matrix* measuring the number of times any pair of objects are in the same cluster for all partitions, consensus clustering is traced back to a clustering problem that makes use of this particular similarity matrix (e.g., [2], [11], [8]). A recent variant of this technique is the use of a co-association matrix of real values, taking into account the size of clusters [6], while a related approach is metaclustering, namely, a clustering of clusterings based on a similarity matrix at the clustering level [12]. Clearly, in all these cases the results depend on the particular clustering algorithm used in the final step.

Perhaps the most popular approach to consensus clustering consists of casting it as an optimization problem, where the objective is to find a consensus partition Π^{opt} such that the sum of the similarities (dissimilarities) between Π^{opt} and q base partitions $\Pi_1, \Pi_2, \ldots, \Pi_q$ is maximal (minimal). Mathematically:

$$\Pi^{opt} = \underset{\Pi}{\operatorname{argmax}} \sum_{i=1}^{q} \operatorname{Sim}(\Pi, \Pi_i), \tag{1}$$

where Sim is a similarity measure between two partitions. This approach is known as finding the *median partition* because the consensus is seen as a median or centroid of the given input partitions. Several types of similarity functions can be used within (1). A common choice is to use functions based on the consensus among the decisions made by the two partitions on individual pairs of objects. This vein of research, originating in the Rand index [13], is now described in detail because it is also the focus of our work.

Given a set O of n objects and two partitions of O to compare, Π_1 and Π_2, the Rand index (RI) is defined as

$$RI = \frac{n_{11} + n_{00}}{n_{00} + n_{01} + n_{10} + n_{11}} = \frac{n_{11} + n_{00}}{\binom{n}{2}}, \tag{2}$$

where

- n_{11}: Number of pairs of objects that are in the same cluster in both Π_1 and Π_2.
- n_{00}: Number of pairs of objects that are in different clusters in both Π_1 and Π_2.
- n_{10}: Number of pairs of objects that are in the same cluster in Π_1 but in different clusters in Π_2.
- n_{01}: Number of pairs of objects that are in different clusters in Π_1 but in the same cluster in Π_2.

Intuitively, one can think of $n_{11} + n_{00}$ as the number of agreements between Π_1 and Π_2, and $n_{10} + n_{01}$ as the number of disagreements between Π_1 and Π_2. The Rand index has a value between 0 (i.e., no agreement on any pair of objects, which happens only when one partition consists of a single cluster while the other consists only of clusters containing single objects) and 1 (i.e., when the two partitions coincide).

As the expected value of the Rand index for random partitions does not take a constant value, it can be corrected for chance. The *adjusted Rand index* [14] assumes the generalized hypergeometric distribution as the model of randomness, i.e., the Π_1 and Π_2 partitions are picked at random such that the number of objects in the clusters and the number of clusters are fixed. The adjusted Rand index takes the value 0 when the index equals its expected value (and is bounded above by 1).[1]

In both the Rand and adjusted Rand index the two types of agreement (i.e., terms n_{11} and n_{00}) have the same importance.

1. Using a similar approach, other clustering similarity measures such as *mutual information* can be corrected for chance [15].

However, in many domains (including web clustering), placing a pair of objects in different clusters is usually not as informative as placing the two objects in the same cluster. It has been argued [16] that pairs of type 00 are not clearly indicative either of similarity or of dissimilarity, as opposed to counts of "good pairs" (term n_{11}) and "bad pairs" (terms n_{10} and n_{01}), thus ending with a formula conceptually similar to Jaccard's coefficient [17]:

$$JC = \frac{n_{11}}{n_{11} + n_{01} + n_{10}}. \quad (3)$$

Another well-known variant of the Rand index is the *Symmetric Difference Distance* (*SDD*), making use only of disagreements. It is defined as

$$SDD = n_{01} + n_{10} = \binom{n}{2} - (n_{11} + n_{00}). \quad (4)$$

In this case, the median partition problem (1) consists of finding the partition with the minimal distance from the base partitions. The median partition problem with the symmetric difference distance is known to be NP-complete, e.g., see [18] and [19]. The SDD measure has been used as the objective function to be optimized in several earlier works on consensus clustering, e.g., [20], [21], including a recent version with weighted input clusterings [22].[2]

In the measures reviewed so far, the individual contribution of the different types of agreements and disagreements to the overall similarity (or dissimilarity) is the same. However, the value of coclustering will likely be higher when the partition contains many clusters and lower with few clusters. Dually, not coclustering will be deemed more (less) valuable with few (many) clusters.

This issue is addressed by our new index (PRI), which is described in the next section. We will see that PRI ensures a better balance of the terms in (2), thus marking the shift from pair-counting to pair-weighting-based consensus functions.

The key features of the main clustering similarity measures based on object-pair analysis are recapitulated in Table 1. In the next section, we show that PRI has all the desirable properties listed in Table 1, except for the lack of a constant expected value, whose utility, however, has been questioned on the grounds of the artificiality of the randomness model [24], and for the unavailability of computational complexity results, which mainly has a theoretical interest.

Other similarity functions that can be used within (1) are the well-known *normalized mutual information* [3], denoted as *NMI*, and a more recent *kernel* method making use of weighted partitions [7]. These two functions are based on an information theoretic analysis at the partition level, rather than on the decisions made on the single object pairs. Due to its popularity and simplicity, in our experiments we will use *NMI* as a term of reference for performance evaluation.

The rationale of *NMI* is that the mutual information $I(X, Y)$ between clusterings X and Y can be normalized by the (geometric) mean of the entropies $H(X)$, $H(Y)$ of the single clusterings, based on the observation that $I(X, Y) \leq \min(H(X), H(Y))$. *NMI* is computed as

$$NMI = \frac{\sum_{i=1}^{m_1} \sum_{j=1}^{m_2} n_{i,j} \log\left(\frac{n \cdot n_{i,j}}{n_i^{(\Pi_1)} n_j^{(\Pi_2)}}\right)}{\sqrt{\sum_{i=1}^{m_1} n_i^{(\Pi_1)} \log\frac{n_i^{(\Pi_1)}}{n} \cdot \sum_{j=1}^{m_2} n_j^{(\Pi_2)} \log\frac{n_j^{(\Pi_2)}}{n}}}, \quad (5)$$

where m_1 and m_2 are, respectively, the number of clusters of Π_1 and Π_2, $n_i^{(\Pi_1)}$ is the number of objects in cluster i according to Π_1, $n_j^{(\Pi_2)}$ is the number of objects in cluster j according to Π_2, and $n_{i,j}$ is the number of objects in cluster i according to Π_1 as well as in cluster j according to Π_2. Note that unlike the standard mutual information, NMI ranges from 0 to 1, and it does not necessarily increase as the number of clusters grows.

Other works do not fit into the framework of (1). The method described in [4] is conceptually similar to a kernel method, but it leads to a different optimization problem. Consensus clustering is seen as a finite mixture of multinomial distributions in the transformed feature space generated by the base partitions. A consensus partition is then found as a solution to the corresponding maximum likelihood problem using the EM algorithm. A similar approach has been taken in [9], using a Bayesian probabilistic mixture model. In both of these probabilistic methods the number of clusters in the consensus partition is prespecified, while it can adapt to the data in a recent refinement of the latter technique [25].

Besides development of better algorithms for creating the combined clustering, recent work has considered related issues such as m-to-m (rather than m-to-1) cluster associations, i.e., mapping m base partitions onto m new partitions which are in closer agreement with one another [26], and cluster ensemble selection, where the goal is to choose a subset of partitions from a given ensemble that achieve improved clustering performance due to their diversity (e.g., [27], [28]). Also, consensus clustering can be formulated as a *correlation clustering* problem, in which the input is a similarity matrix of objects (rather than a set of partitions of the objects) and the objective is to find a partition that

TABLE 1
Features of Clustering Similarity Measures Based on Object-Pair Analysis

	RI	*SDD*	*JC*	*ARI*	*PRI*
Agreements & disagreements	x		x	x	x
Variable # of clusters	x	x	x		x
Variable # of cluster objects	x	x	x		x
Prob. of object-cluster assignment					x
Bounded function	x	x	x		x
Constant expected value				x	
Theoretical complexity		x			

RI = Rand index, SDD = symmetric difference distance, JC = Jaccard's coefficient, ARI = adjusted Rand index, PRI = probabilistic Rand index.

2. The last variant of the Rand index that we are aware of is the *probabilistic Rand index* presented in [23]. Despite its name, it is mostly unrelated to our work. It has a narrow scope and is suitable for very specific applications, e.g., handling refinements of clusters in the context of image segmentation.

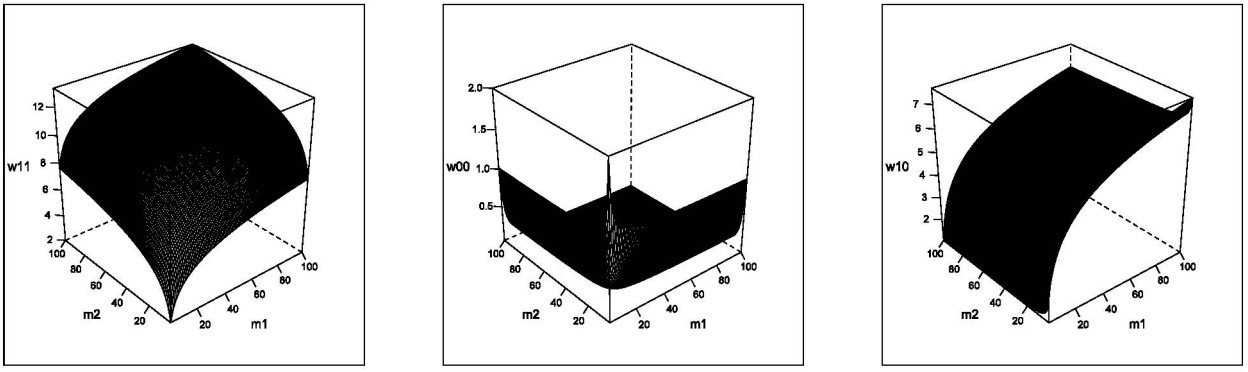

Fig. 1. Value of PRI weights as a function of m_1 and m_2, for $n = 100$.

maximizes intracluster similarity (or, equivalently, the number of agreements) and minimizes intercluster similarity (or, equivalently, the number of disagreements), e.g., [29], [21], [30].

3 Probabilistic Rand Index

3.1 Index Definition

Rather than counting the different types of agreements and disagreements, it may be more convenient to weigh such events on the basis of the probability of their occurring by chance. The rationale is that as the number of clusters grows, the chance for a pair of objects to be placed in the same cluster decreases, while at the same time the chance to be placed in different clusters increases. We can estimate the relative importance of terms $n_{00}, n_{01}, n_{10}, n_{11}$ as a function of the number of clusters m_1 and m_2 in partitions Π_1 and Π_2, respectively.

Assuming that each object is randomly assigned to one cluster, the probability that two objects are in the same cluster in both partitions is

$$p_{11} = \frac{1}{m_1} \cdot \frac{1}{m_2}, \tag{6}$$

the probability that two objects are in different clusters in both partitions is

$$p_{00} = \frac{m_1 - 1}{m_1} \cdot \frac{m_2 - 1}{m_2}, \tag{7}$$

the probability that two objects are in the same cluster in Π_1 and in different clusters in Π_2 is

$$p_{10} = \frac{1}{m_1} \cdot \frac{m_2 - 1}{m_2}, \tag{8}$$

and the probability that two objects are in different clusters in Π_1 and in the same cluster in Π_2 is

$$p_{01} = \frac{m_1 - 1}{m_1} \cdot \frac{1}{m_2}, \tag{9}$$

with $\sum_{h=00}^{11} p_h = 1$ (expressing h in binary). The lower the probability, the higher the information content associated with the observation that the event indeed occurred. We estimate the weights associated with each of the four possible types of agreements or disagreements with the *self information*, i.e:

$$w_h = -\log_2 (p_h). \tag{10}$$

Such weights can be used to define a Probabilistic Rand Index:

$$PRI = \frac{w_{11} \cdot n_{11} + w_{00} \cdot n_{00}}{w_{00} \cdot n_{00} + w_{01} \cdot n_{01} + w_{10} \cdot n_{10} + w_{11} \cdot n_{11}}. \tag{11}$$

PRI is a truly probabilistic version of the Rand index. Analogously to RI, PRI varies between 0 (i.e., no agreement on any pair of objects) and 1 (i.e., when the two partitions are equal). Also, the similarity between two partitions is still expressed as a ratio of agreements on object-pair groupings, which is a simple and natural interpretation of clustering similarity. The advantage of using PRI over RI is that the different types of agreements and disagreements are weighted with their significance.

As an illustration, consider the following two partitions of 10 objects into four clusters:

$$\Pi_1 = (\text{A B C D}) (\text{E F G}) (\text{H I}) (\text{J}),$$
$$\Pi_2 = (\text{A E H J}) (\text{B F I}) (\text{C G}) (\text{D}).$$

The two partitions do not look similar as they never place two objects in the same cluster (i.e., no agreements of type 11). However, they have a relatively high Rand index value (i.e., $RI = 0.555$) because there are 25 agreements of type 00. By contrast, their probabilistic Rand index is much lower, i.e., $PRI = 0.300$. Now consider the partition Π_2' obtained from Π_2 by moving objects B and C into the first cluster:

$$\Pi_2' = (\text{A B C E H J}) (\text{F I}) (\text{G}) (\text{D}).$$

Partition Π_2' more closely resembles Π_1 than Π_2 does because three pairs of objects have been coclustered (i.e., A-B, A-C, B-C). However, the RI value stays the same as the total number of agreements between Π_1 and Π_2' is still 25, namely, three agreements of type 11 and 22 of type 00. In contrast, the PRI rises from 0.300 to 0.385.

3.2 Analysis of PRI Weights

In this section, we study the variation of the PRI weights, defined in (6) through (10), as a function of m_1, m_2, which in turns are bounded by the number of objects n. In Fig. 1, we plotted the surfaces w_{11} (left), w_{00} (middle), and w_{10} (right), corresponding to all possible pairs of values m_1, m_2, in the range between 1 and 100.

The value of w_{11} grows monotonically with m_1 and m_2, varying from 0, for $m_1 = m_2 = 1$, to $2\log_2(n-1)$, for

TABLE 2
Value of PRI Weights as a Function of m, with $m_1 = m_2 = m$

	2	3	4	5	10	50	100
w_{11}	2.00	3.17	4.00	4.64	6.64	11.29	13.29
w_{00}	2.00	1.17	0.83	0.64	0.30	0.06	0.03
w_{10}=w_{01}	2.00	2.17	2.42	2.64	3.47	5.67	6.66

$m_1 = m_2 = n - 1$.[3] By contrast, the function w_{00} decreases monotonically with m_1 and m_2. Its maximum is equal to 2, for $m_1 = m_2 = 2$,[4] while its minimum is $2\log_2 \frac{n-1}{n-2}$, achieved for $m_1 = m_2 = n - 1$.[5] The weight w_{10} increases with m_1 (m_2 being constant) and decreases with m_2 (m_1 being constant). It achieves its maximum (i.e., $\log_2 2n$) for $m_1 = 100$, $m_2 = 2$.[6] The behavior of w_{01} is dual to that of w_{10}.

The relative importance of the four weights in the computation of PRI can be better understood by looking at Table 2, where we report their value for a fixed choice of the number of clusters, assuming $m_1 = m_2 = m$. The figures show that in realistic situations, agreements of type 11 receive twice as much credit as that of disagreements, with the weight of agreements of type 00 being much smaller. For instance, for $m = 10$ (as in the information retrieval experiments reported in this paper), we get $w_{11} = 6.64$, $w_{10} = w_{01} = 3.47$, and $w_{00} = 0.30$.

4 Optimizing the PRI Value of a Clustering Ensemble

4.1 Problem Definition

Based on PRI, a pairwise index of partition similarity, we now define a similarity measure between a single partition Π and a set of q partitions $\Pi_1, \Pi_2, \ldots, \Pi_q$ as

$$\Psi^{(PRI)}(\Pi, \Pi_1, \Pi_2, \ldots, \Pi_q) = \frac{1}{q}\sum_{r=1}^{q} PRI\ (\Pi, \Pi_r). \tag{12}$$

This is our objective function. The optimal partition Π^{opt} is the one that has maximal similarity with the given partitions:

$$\Pi^{opt} = \arg\max_{\Pi}\ \Psi^{(PRI)}(\Pi, \Pi_r). \tag{13}$$

The optimization of $\Psi^{(PRI)}$ is computationally expensive, due to the huge size of the search space. The number N_m of distinct partitions of n objects into m nonempty clusters is [32]:

$$N_m = \frac{1}{m!}\sum_{i=0}^{m}(-1)^{m-i}\binom{m}{i}i^n. \tag{14}$$

When the number of clusters may vary from 1 to n, as in our case, the number N of distinct partitions grows to

$$N = \sum_{j=1}^{n}\frac{1}{j!}\sum_{i=0}^{j}(-1)^{j-i}\binom{j}{i}i^n. \tag{15}$$

Consider again the partitions Π_1 and Π_2 introduced in Section 3.1, seen as two input partitions to be optimized. Using (15), the number N of distinct partitions of 10 objects is 115,975. We generated all possible partitions and computed the $\Psi^{(PRI)}$ score associated with each, seen as a consensus clustering of Π_1 and Π_2. We found four optimal partitions, namely, (A B C D) (E F G) (H I J), (A B C D) (E H J) (F G I), and, symmetrically, (A E H J) (B F I) (C D G) and (A E H J) (B C D) (F G I), with $\Psi^{(PRI)} = 0.662$. The optimal partitions have three clusters, rather than four. Intuitively, the consensus strategy maintained or swapped the two largest clusters in each input partition, while at the same time merging the two smallest clusters. The latter operation has the main effect of increasing the contribution made by the agreements of type 00 to the final $\Psi^{(PRI)}$ value because, with three clusters instead of four, it is significantly more difficult for a pair of objects not to be coclustered in both partitions. The value of n_{00} slightly decreases (from 60 to 57), but w_{00} increases from 0.83 (with $m_1 = m_2 = 4$) to 1 (with $m_1 = 3$, $m_2 = 4$), with the value of $w_{00} \cdot n_{00}$ increasing from 49 to 57.

Note that if we had chosen one of the original clusterings as consensus clustering, we would have obtained the same $\Psi^{(PRI)}$ value for each (i.e., $\Psi^{(PRI)} = 0.650$).[7] Clearly, in this example the improvement attainable by the optimal $\Psi^{(PRI)}$ value is very limited because, with few clusters and few objects, choosing one of the two original partitions as consensus clustering ensures a fair similarity with the other partitions (including itself). Should we choose to optimize the RI instead of the PRI value, the best partition would coincide with either input partition, with $\Psi^{(RI)} = 0.777$. In this case, the optimal partition has a total of 80 agreements with Π_1 and Π_2, more than those associated with each optimal PRI partition (i.e., 78).

Clearly, brute force methods cannot be applied to find Π^{opt} in real applications. For instance, with 100 objects (i.e., the typical number of search results fetched by web clustering engines), we get $\approx 10^{39}$ distinct partitions. In the next section we study approximate solutions to this problem.

4.2 An Approximate Solution to the Optimization Problem

Several heuristic solutions have been devised to find Π^{opt} in (1) [5], including mapping the problem onto a graph-based partitioning setting [3], using genetic algorithms [33], or using sophisticated metaheuristic strategies such as simulated and quantum annealing [31]. However, these methods are computationally expensive and do not support, in general, real-time consensus clustering. We used a simple, efficient stochastic hill-climbing strategy. It consisted of starting with a partition and iteratively moving a single object to a different (possibly empty) cluster until the objective function associated with the newly created partition increases. In our implementation, the initial partition was the input partition with the highest similarity value with the given partitions. We then selected any successor randomly, and halted the computation after

3. w_{11} is not defined when m_1 or m_2 are equal to n because in this case there cannot be agreements of type 11.

4. w_{00} is not defined when m_1 or m_2 are equal to 1 because in this case there cannot be agreements of type 00.

5. Note that (7) does not hold when there is a degenerate partition containing n clusters with one object. In this case, p_{00} is equal to $\frac{m-1}{m}$ if the other partition contains $m(< n)$ clusters, or to 1 if $m_1 = m_2 = n$.

6. w_{10} is not defined for $m_1 = 1$ because in this case there are no agreements of type 10.

7. This holds for any pair of partitions because PRI is symmetric and it is equal to 1 when the two partitions are equal. We get $\Psi^{(PRI)}(\Pi_1, \Pi_1, \Pi_2) = [PRI(\Pi_1, \Pi_1) + PRI(\Pi_1, \Pi_2)]/2 = [PRI(\Pi_2, \Pi_2) + PRI(\Pi_2, \Pi_1)]/2 = \Psi^{(PRI)}(\Pi_2, \Pi_1, \Pi_2)$.

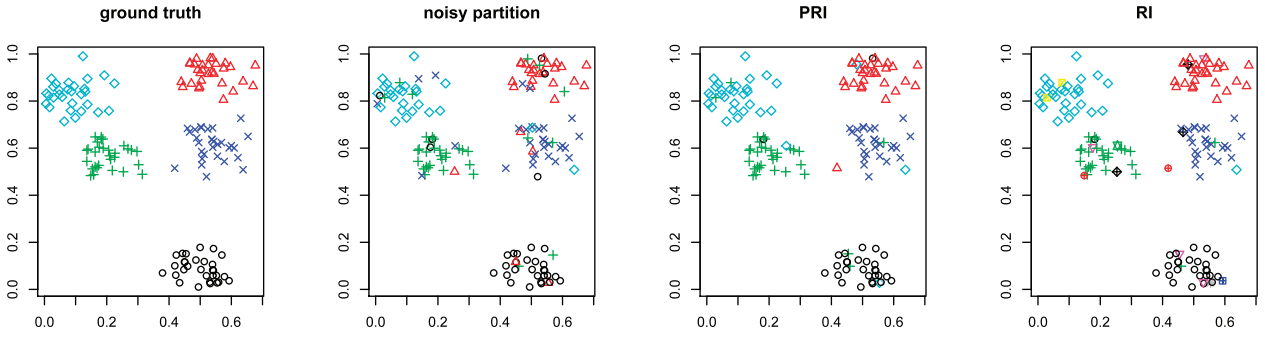

Fig. 2. True classes of our synthetic dataset and three automatically generated partitions, one obtained by injecting random noise and the other two by combining the noisy partitions.

testing all possible nm successors of the current partition. We chose this termination criterion partly because the space of moves involving single objects is sufficiently small that it can be explored exhaustively, and partly because we experimentally checked that using more CPU-intensive methods such as random-restart hill climbing did not produce better results.

4.3 An Example

As an illustration of the full method, we now describe an example of the potential of PRI-based consensus clustering for combining the results of noisy partitions. A set of synthetic data was created using the following procedure. Five fictitious centroids were selected at random in the unit square, and 30 points were generated from the normal distribution around each centroid. The result is shown in the first chart of Fig. 2. We then randomly added noise to the ground-truth partition by randomly modifying a fixed percentage (i.e., 10, 20, and 30 percent) of the cluster-object assignments. In each case, we generated three base noisy partitions. One of the base partitions with 20 percent of errors is shown in Fig. 2 (second chart from the left). The three base partitions were then combined using both the PRI-based and the RI-based consensus clustering methods. The optimal partitions found with 20 percent of noise are plotted in Fig. 2 (third and fourth charts from the left, respectively, for PRI and RI).[8]

The whole procedure was iterated 10 times, using different sets of centroids. In Table 3, we show the number of tested candidate partitions and of clusters and errors of the final partition, with results averaged over the multiple runs, for each amount of noise and each method. By exploring a very small portion of the search space, the PRI-based method achieved much better accuracy than base partitions for all values of noise.[9] By contrast, RI was slower and made more errors than PRI. The growth in the number of tested candidates was probably related to the difficulty of generating an improved partition from base partitions with a higher consensus value compared to PRI. RI was also less accurate than PRI, mainly because the final consensus clusterings generated by RI contained too many clusters. We checked that the RI-based partitions contained all misclassifications of the corresponding PRI-based partitions and, in addition, other errors due to placing some objects into newly-created clusters. Fig. 2, for instance, clearly shows that the RI-based partition contains several clusters with just one or two data points. This behavior can be explained considering that moving an object into a new cluster has, in general, a positive effect on the RI value due to an increase in the number of agreements of type 00, whereas using PRI this effect is counterbalanced by their downweighting.

We also studied how the performance of PRI-based consensus clustering varies as the number of base partitions increases. We generated 10 partitions with 20 percent of noise and then iteratively computed the consensus partition from $2, 3, \ldots, 10$ of them. The number of errors monotonically decreased from 30 (with two base partitions the consensus partition was equal to either of the two) to 1 (using 10 base partitions). Symmetrically, the improvement of the PRI value of the consensus partition over that of the best initial partition monotonically increased, mainly because the score of the latter became smaller as more input partitions were used.

5 Experimental Evaluation

5.1 Nonconvex Clusters

Artificial datasets are often used to illustrate how a suitable combination can reveal underlying nonconvex clusters which cannot be found by the base partitions, usually generated by the k-means clustering method. One well-known example is the Cassini dataset, containing 1,000 2D points split into three classes, two outer banana-shaped classes and a circle in the middle.[10] For this experiment, we used the R package *mlbench*, providing a computational environment for creating and analyzing the partitions generated by k-means on several datasets including Cassini.

In most work on combining multiple k-means partitions the base partitions are generated by applying the same k-means algorithm with different initializations. This approach has the disadvantage that the experiments may not be easily replicable and that the partitions generated for the Cassini dataset by changing the initial k-means seeds are often almost identical. In our experiments, we used a different

8. Note that the optimal partitions do not depend on the locations of the points, i.e., if an alternative method were used to place the points in R2, the results would be identical.

9. The number of errors made by base partitions is 15, 30, and 45, for 10, 20, and 30 percent of noise, respectively.

10. It can be proven [34] that the true Cassini partition is only a local minimum of k-means error function, while the global minimum of k-means is visually similar to the output produced by Lloyd in Fig. 3 and has approximately 77 percent of correct decisions.

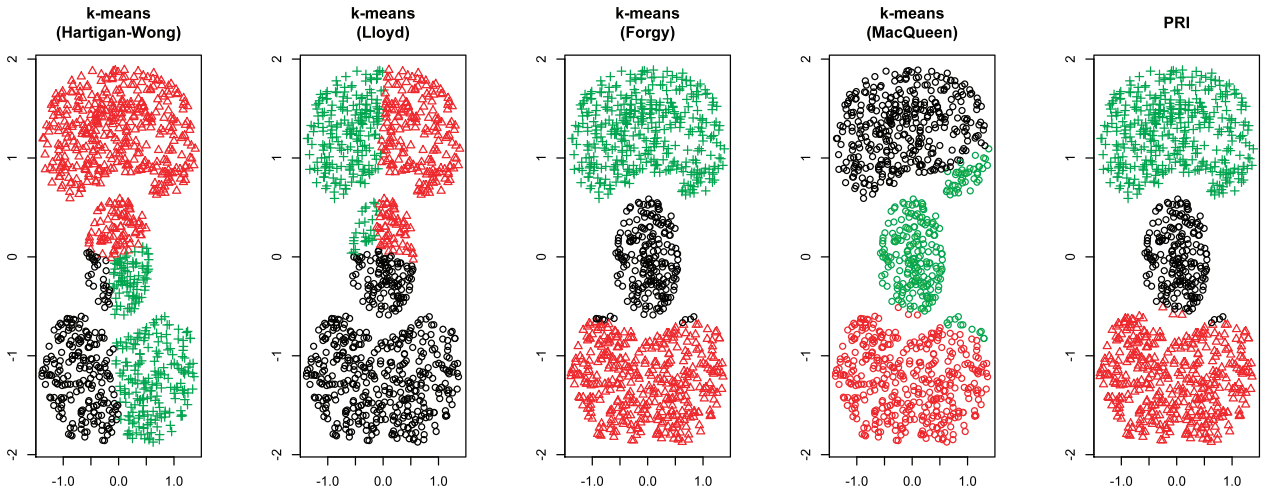

Fig. 3. Five automatically generated partitions of the Cassini dataset, using four variants of k-means and PRI-based consensus clustering.

approach. We considered four algorithmic variants of k-means made available in R, namely, Lloyd, MacQueen, Hartigan-Wong, and Forgy. The initial partitions used in our experiments can be easily reproduced through the following R script:

```
# # Create Cassini
library("mlbench")
set.seed(1234)
Cassini <- mlbench.cassini(1000)
set.seed(3)
# # Create the Lloyd k-means partition
partyLloyd <- kmeans(Cassini$x, 3, 10, 1, "Lloyd")
...
```

where the arguments 3, 10, and 1 are, respectively, the number of clusters, iterations, and random sets. As shown in Fig. 3, the four partitions were very different, with varying resemblance to the true Cassini classes. The closest one is Forgy, in which only 11 data points at the edges were misclassified.

We then computed the RI, NMI, and PRI consensus partitions, starting from the Forgy partition and trying to improve the corresponding objective functions in (1) using the approximate procedure described in Section 4.2.[11] In this case, the number of clusters in the final consensus partition was set to three for all ensemble methods, thus restricting the heuristic search to the partitions with exactly three clusters.

The best result was obtained by PRI, shown in the rightmost plot in Fig. 3. The number of errors decreased from 11 to 7. On close inspection, we found that in the 10 iterations required to find the local maximum, seven objects misclassified by Forgy (i.e., the black ones in the left part of the red banana) were correctly reassigned to the right cluster by PRI, while three correctly classified objects in Forgy were moved to a wrong cluster in PRI (i.e., the red objects in the black circle). We also found that RI too improved over Forgy, although by a smaller extent than PRI, whereas NMI was not able to find a better partition than Forgy, i.e., its output was exactly the same as Forgy.

In order to get more reliable results, we repeated this experiment multiple times, changing the values of the parameters used to generate the four k-means base partitions. We kept the number of clusters constant (i.e., equal to three), but we let the value of random sets vary from 1 to 3 [12] and that of iterations from 5 to 15. The average number of errors of the individual and consensus clustering methods are shown in Table 4. These findings substantially agree with those shown in Fig. 3.

It is also interesting to note that the absolute performance achieved by PRI compares favorably with the results reported in the literature, especially in light of its high computational efficiency. For instance, in [7], where the Cassini data points are first mapped into a 1D feature space, the best consensus clustering method makes 14 errors. A better, near-perfect classification may be achieved, but at the cost of increased complexity, e.g., by repeatedly training on resampled sets [34] or by employing multiple clustering algorithms in addition to k-means to form the base partitions [35].

TABLE 3
Performance of PRI-Based and RI-Based Consensus Clustering Methods on the Noisy Synthetic Dataset for Different Values of Noise

	PRI (10%)	RI (10%)	PRI (20%)	RI (20%)	PRI (30%)	RI (30%)
Tests	4557	12356	10867	55521	5192	172753
Clusters	5	8	5	12	5	16
Errors	7	8	12	18	33	45

TABLE 4
Average Number of Errors of k-Means and Consensus Clustering Methods on the Cassini Dataset

K1	K2	K3	K4	RI	NMI	PRI
325	358	58	18	11	16	9

The four variants of k-means are $K1 = Lloyd$, $K2 = Hartigan\text{-}Wong$, $K3 = MacQueen$, *and* $K4 = Forgy$.

11. Note that the consensus partitions of the Rand index and SDD methods coincide.

12. We checked that, with a higher number of random starts, the four k-means algorithms produce nearly identical results, thus reducing the diversity of input partitions and the scope for their combination.

5.2 Subtopic Retrieval with Web Search Results Consensus Clustering

Conventional web search engines do not adequately deal with broad or ambiguous queries because they provide a flat list of search results. Web Search Results Clustering (WSRC) systems organize search results by topic, where each topical cluster is described by a textual label. Then they give the user the possibility of seeing the results associated with any cluster by clicking on its label, thus allowing, in principle, direct access to the pages pertaining to distinct interpretations (or subtopics) of the given query.

In contrast to conventional clustering, WSRC algorithms try to optimize not only the clustering structure but also the quality of cluster labels because a cluster with a poor description is very likely to be entirely omitted by the user, even if it points to a group of strongly related and relevant search results. In the last 10 years, a number of different WSRC algorithms with improved usability have been developed, including several commercial web clustering engines.[13] This relatively rich body of literature is surveyed in [10]. Because there are so many available WSRC systems, it is tempting to combine their outputs, just as the results of several search engines can be merged into a meta search engine. To the best of our knowledge, this problem has not been addressed so far, except for [31].[14]

In our experiments we used three state-of-the-art WSRC systems, namely, Lingo [36], KeySRC [37], and Lingo3G.[15] All these clustering algorithms are characterized by highly descriptive phrases as cluster labels, and are known to perform well on browsing retrieval tasks [10]. In Lingo, frequent phrases are extracted from search results using suffix arrays [38], then the frequent phrases that best match certain *latent topics* present in the search results, determined via *singular value decomposition* [39], are selected, and finally documents are allocated to such frequent phrases. KeySRC generates clusters labeled by *keyphrases*. The keyphrases are extracted from the generalized suffix tree [40] built from the search results and merged through an improved hierarchical agglomerative clustering procedure, representing each phrase as a weighted document vector and making use of a variable dendrogram cut-off value. Lingo3G is a commercial system developed by Carrot Search. Despite similar names, Lingo and Lingo3G are two very different clustering algorithms. Lingo3G employs a custom-built metaheuristic algorithm that aims to select well-formed and diverse cluster labels.

As a test collection, we used AMBIENT[16] [41], explicitly designed for evaluating the subtopic retrieval effectiveness of systems that postprocess search results. It has 44 queries extracted from the *ambiguous* Wikipedia entries, each with a variable number of interpretations and a list of 100 ranked search results annotated with interpretation relevance judgments. To evaluate the performance of WSRC systems, we used the *Subtopic Search Length under k document sufficiency* ($kSSL$), introduced in [37]. It is defined as the average number of items (cluster labels or search results) that must be examined before finding a sufficient number (k) of documents relevant to any of the query's subtopics, assuming that both cluster labels and search results are read sequentially from top to bottom, and that only cluster with labels relevant to the subtopic at hand are opened. Clearly, the shorter the $kSSL$ value, the better the retrieval effectiveness.

We ran the three WSRC systems being tested on the 100 search results associated with each AMBIENT query and evaluated the performance of the corresponding output partitions using $kSSL$, with $k = 1, 2, 3, 4$.[17] Then for each query we found the consensus partition generated from the three individual partitions by three clustering ensemble methods, namely, RI, NMI, and PRI.

We next computed the cluster labels of each consensus partition. As we cannot use available techniques for cluster labeling because of our assumption that the descriptions of the objects are not available, the idea is to select the most agreed-upon labels from those returned by the individual clustering algorithms. The consensus labeling algorithm consists of the following three steps:

1. We associate with each cluster of the consensus partition a set of candidate labels, formed by all labels under which each search result in the cluster has been classified in at least one individual method.
2. We assign a score to each candidate label based on both its *extensional* coverage of the set of objects and *intensional* coverage of the set of labels. The purpose of the intensional factor is to promote syntactically different labels that refer to the same concept. The exact formula is the following:

$$Score(l) \;=\; count(obj) \cdot \sum_{w \in l} count(w), \qquad (16)$$

   where $count(obj)$ is the number of search results in the cluster that are labeled by l, w is a nonstop word contained in l, and $count(w)$ is the number of distinct labels in the cluster that contain word w.
3. We select the label with the highest score.

An example is given in Table 5, where we show the set of cluster labels generated by the three input algorithms and by PRI that were judged to be relevant to at least one of the interpretations of the query "Jaguar" defined in the AMBIENT collection. Finally, we found the $kSSL$ values of the clustering ensemble methods enriched with the cluster labels.

The overall results, averaged over the set of queries, are reported in Table 6. Since the algorithm for finding the optimal partition is approximated, the results of each ensemble method were obtained as an average over 10 runs, changing the random order of candidate partitions.[18] For further comparison, we also included the $kSSL$ value of the

13. A well-known example is Yippy at http://search.yippy.com (formerly Clusty).

14. The research described here builds in part on [31]. The information retrieval experimental setting was the same. The cluster similarity measure had the same inspiration of weighing agreements and disagreements but it was computed as an algebraic sum, thus resulting in an altogether different index.

15. http://carrotsearch.com/lingo3g-overview.html.

16. http://credo.fub.it/ambient.

17. We considered only the AMBIENT interpretations with at least two search results because the tested systems, like most web clustering engines, do not generate singleton clusters.

18. Different runs often yielded an identical consensus partition. This observation was confirmed in the experiments described in the next section, using a different dataset.

TABLE 5
Cluster Labels Relevant to the AMBIENT Interpretations for the Query "Jaguar"

Jaguar query	PRI	KeySRC	Lingo	Lingo3G
Interpr1	jaguar panthera onca / big cats	jaguar panthera onca / big cats	jaguar panthera onca / big cats	jaguar panthera onca / big cats
Interpr2	jaguar models / ford motor company	jaguar models / jaguar parts / ford motor company	jaguar models / jaguar parts / ford motor company	jaguar models / jaguar parts
Interpr3	atari jaguar	atari jaguar	atari jaguar	
Interpr4	mac os x	mac os	mac os	mac os x

The interpretation definitions are the following. Subtop1: "Jaguar (Panthera onca), a New World mammal (a "big cat") of the Felidae family native to South and Central America"; Subtop2: "Jaguar (car), a British luxury car manufacturer, owned by Ford as of 1990"; Subtop3: "Atari Jaguar, a video game console made by Atari"; Subtop4: "Jaguar, the codename for Mac OS X v10.2."

plain search engine. The most effective result of the actual clustering methods for each value of k is reported in bold.

Table 6 shows that PRI obtained better results than any individual and ensemble method for all evaluation measures. The performance improvement was marked over both the individual methods, ranging from 4.9 percent (over Lingo for $k = 2$) to 21.6 percent (over Lingo3G for $k = 3$), and the ensemble methods, ranging from 7.6 percent (over RI for $k = 2$) to 14.5 percent (over NMI for $k = 1$). All the differences were statistically significant, using a two-tailed paired t test with a confidence level in excess of 95 percent. Unlike all individual and ensemble methods, PRI clearly improved on the search engine baseline not only for $k \geq 2$ but also for $k = 1$.

Looking at the performance of ensemble methods, the PRI strategy of forming (nonsingleton) clusters only when finding enough probabilistic evidence was apparently able to reduce the number of irrelevant or redundant clusters. A query-by-query analysis revealed that both RI and NMI were characterized by more numerous clusters per query, but this did not result in better query interpretation coverage due to cluster redundancy. At the same time, a greater search length overhead was incurred by these two systems when a certain interpretation was not covered by the cluster labels and it was necessary to switch to the whole search results list. By contrast, we observed that PRI produced fewer yet discriminative clusters for queries with many interpretations, while it avoided generating redundant clusters on queries with few interpretations.

Besides retrieval performance, we compared the running speed of the three ensemble methods. The overall processing time to find a consensus partition is given by the product of the number of tested partitions and the time necessary to score each of them. For RI and PRI, the latter is proportional to the number of objects n since, in order to compute the consensus value associated with a new partition, it suffices to reevaluate the (n-1) pairs containing the object being moved (rather than all possible pairs of objects). In Table 7, we report the average number of tests and the average running times (on a computer of medium power, i.e., 2.8 Ghz CPU, 4 GB RAM) for RI, NMI, and PRI , using AMBIENT and ODP-239, an additional test collection that will be described in the next section. AMBIENT and ODP-239, have different features but are both characterized by multiple datasets containing about 100 objects (i.e., web search results) each, thus with a large partition search space. The results of each clustering consensus method were again averaged over 10 runs.

PRI required fewer tests and converged faster than RI and NMI across both collections. These findings confirm the results obtained on the noisy dataset in Section 4.3. It is also worthy of note that the times of RI and PRI are proportional with a similar factor to the number of tests since the PRI weights are constant and thus the cost of computing the PRI similarity between two partitions is the same as RI. Table 7 clearly suggests that the PRI method is suitable for real-time search applications because its processing times were on the order of tens of milliseconds.

TABLE 6
Retrieval Performance on the AMBIENT Test Collection Measured as Mean kSSL over the Set of Queries, for Several Values of k

	kSSL (k=1)	kSSL (k=2)	kSSL (k=3)	kSSL (k=4)
Individual methods				
KeySRC	14.4	24.3	31.6	36.8
Lingo	15.0	24.1	31.1	36.4
Lingo3G	15.8	27.0	36.0	40.6
Search engine	14.3	31.1	40.0	47.3
Ensemble methods				
RI	14.7	24.8	30.9	36.3
NMI	15.1	25.6	31.5	36.7
PRI	**12.9**	**22.9**	**28.2**	**33.4**

5.3 Recovering the Topics of ODP Web Pages

In this section, we evaluate the ability of the clustering systems to recover the topical structure of web pages. From a conceptual point of view this is the same task as addressed in the Cassini experiments, i.e., recovering known classes from a gold standard partition. The main differences are that we use web instead of synthetic data, and that we apply the algorithms to a large number of datasets, one for each web topic. Unlike the subtopic retrieval experiments, we are only interested in the groups generated by the clustering algorithm, regardless of their labels or their usage for interactive search.

TABLE 7
Number of Candidate Partitions and Running Times of RI, NMI, and PRI on AMBIENT and ODP-239, Averaged over the Set of Topics and Multiple Runs

	RI	NMI	PRI
AMBIENT			
Tests	30,367	35,835	19,118
Time (msecs)	108	256	66
ODP-239			
Tests	118,914	57,478	12,432
Time (msecs)	404	405	41

TABLE 8
Subcategories of ODP-239 Topic 221: "Sports > Cycling"

221.1	Sports > Cycling > Organizations (22)
221.2	Sports > Cycling > Travel (21)
221.3	Sports > Cycling > Bike_Shops (18)
221.4	Sports > Cycling > Regional (10)
221.5	Sports > Cycling > Mountain_Biking (10)
221.6	Sports > Cycling > Racing (7)
221.7	Sports > Cycling > BMX (5)
221.8	Sports > Cycling > Human_Power_Vehicles (4)
221.9	Sports > Cycling > College_and_Univ (4)
221.10	Sports > Cycling > Personal_Pages (4)

The number of snippets associated with each subcategory is shown in parentheses.

TABLE 9
Classification Performance on the ODP-239 Collection Measured as Mean (Microaveraged) F_β over the Set of Topics, for Several Values of β

	F_1	F_2	F_5
Individual methods			
KeySRC	0,29	0.31	0.34
Lingo	0.27	0.28	0.29
Lingo3G	0.31	0.29	0.28
Ensemble methods			
RI	0.29	0.29	0.29
NMI	0.28	0.28	0.27
PRI	**0.32**	**0.33**	**0.36**

The input clustering algorithms were the same as in the subtopic retrieval section. As a test collection we used ODP-239,[19] first introduced and briefly described in [31]. ODP-239 combines the features of search results data with those of classification benchmarks. It consists of 239 topics, each with about 10 subtopics and 100 page snippets. The topics, subtopics, and their associated snippets were selected from the Open Directory Project[20] data as of June 2009 in such a way that all ODP second-level categories were represented (except for the children of the "Regional" and "World" top categories because they were not truly thematic subcategories), with the distribution of snippets across subtopics reflecting the relative importance of subtopics. The snippets were the items contained in the leaf nodes of the directory. Each snippet consists of a title (which is the anchor text to the final web page pointed to by ODP) and a short textual description. An example illustrating an ODP-239 topic with its associated subtopics is given in Table 8.

In order to assess how good the clustering methods were at recovering the ODP subcategories (referred to as classes), it is convenient to use the well-known F_β *measure* [42], thoroughly discussed in [43], together with related text cluster validity metrics. The F_β *measure* combines precision P and recall R:

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P + R} \tag{17}$$

with

$$P = \frac{TP}{TP+FP},\ R = \frac{TP}{TP+FN}, \tag{18}$$

where TP, FP, and FN are, respectively, the number of true-positives (i.e., two objects of the same class assigned to the same cluster), false-positives (i.e., two objects of different classes assigned to the same cluster), and false-negatives (i.e., two objects of the same class assigned to different clusters). The parameter β is a weighting factor for the importance of the recall (or precision). Because in the web clustering domain false negatives are more harmful than false positives, we may want to give more weight to recall.

We ran Lingo, KeySRC, and Lingo3G on the 239 datasets of ODP-239, and then computed the consensus partitions generated by RI, NMI, and PRI from the individual partitions on each dataset. In Table 9, we report the mean (microaveraged) F_β values for each clustering method, with $\beta = 1, 2, 5$. The results of each ensemble method were again averaged over 10 runs.

Table 9 shows that PRI clearly outperformed all individual and ensemble methods for all evaluation measures, with higher values of β leading to greater performance improvements. Consistent with the results found for the AMBIENT dataset, the performance of RI was, in general, slightly higher than NMI. It may be conjectured that, by assigning smaller weights to agreement of type 00, PRI was effectively able to reduce the false-negatives without sacrificing the true-positives. Note also that although PRI was a clear improvement over the other methods, its performance is, on an absolute scale, still relatively low. This is probably due to the intrinsic difficulty of the classification task on the ODP-239 collection, where texts are very short and subtopics do not always have very distinct meanings and/or discriminative words.

5.4 Validity and Scope of Results

At the end of this evaluation section, we would like to make a general remark about the validity of our findings. The results obtained in the three preceding experimental tasks must be taken with caution because the optimal partitions were computed using the heuristic method described in Section 4.2. While the same approximate optimization method (i.e., stochastic hill climbing) was applied to all ensemble strategies, it is conceivable that a different optimization method would yield different results.

It is also worth noting that in our experiments, consistent with the assumptions of the overall method, the base clustering algorithms worked on the same set of search results. This is not a problem if such algorithms run locally, but it may prevent truly distributed consensus clustering. One direction for future work is to extend the proposed framework to partitions of different but overlapping sets of objects. This extension would open new interesting applications of consensus clustering, such as creating a metaclustering engine that gathers its inputs from base clustering engines fetching their search results from distinct search engines. This is the typical situation encountered when fetching clustered search results from web resources. It applies not only to web pages, but also to blogs, news, or twitters.

6 Conclusion

We presented a new consensus clustering method that is theoretically well founded, computationally efficient, and remarkably effective. The key to this method is a new

19. http://credo.fub.it/odp239.
20. http://www.dmoz.org.

measure for comparing two partitions, called Probabilistic Rand Index, that extends earlier Rand-style indices from object-pair counting to object-pair weighting. We showed that a simple stochastic optimization algorithm delivers improved consensus partitions fast. Across a range of evaluation techniques, the proposed method achieved notable performance improvements over the base clustering algorithms as well as over alternative consensus clustering algorithms based on finding the median partition.

ACKNOWLEDGMENTS

The authors are grateful to two anonymous reviewers for their invaluable comments and suggestions. They would also like to thank Massimiliano D'Amico for providing the output of KeySRC in an XML format, and Stanislaw Osiński and Dawid Weiss for making the results of Lingo and Lingo3G on AMBIENT and ODP-239 available to them.

REFERENCES

[1] A.K. Jain, "Data Clustering: 50 Years beyond K-Means," *Pattern Recognition Letters,* vol. 31, no. 8, pp. 651-666, 2010.
[2] A. Fred and A. Jain, "Data Clustering Using Evidence Accumulation," *Proc. 16th Int'l Conf. Pattern Recognition,* pp. 276-280, 2002.
[3] A. Strehl and J. Ghosh, "Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions," *J. Machine Learning Research,* vol. 3, pp. 583-617, 2002.
[4] A. Topchy, A.K. Jain, and W. Punch, "A Mixture Model for Clustering Ensembles," *Proc. SIAM Int'l Conf. Data Mining,* pp. 379-390, 2004.
[5] A. Goder and V. Filkov, "Consensus Clustering Algorithms: Comparison and Refinement," *Proc. Ninth Workshop Algorithm Eng. and Experiments,* pp. 109-117, 2008.
[6] X. Wang, C. Yang, and J. Zhou, "Clustering Aggregation by Probability Accumulation," *Pattern Recognition,* vol. 45, no. 2, pp. 668-675, 2009.
[7] S. Vega-Pons, J. Correa-Morris, and J. Ruiz-Shulcloper, "Weighted Partition Consensus via Kernels," *Pattern Recognition,* vol. 43, no. 8, pp. 2712-2724, 2010.
[8] A. Topchy, A.K. Jain, and W. Punc, "Clustering Ensembles: Models of Consensus and Weak Partitions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 12, pp. 1866-1881, Dec. 2005.
[9] H. Wang, H. Shan, and A. Banerjee, "Bayesian Cluster Ensembles," *Proc. SIAM Int'l Conf. Data Mining,* pp. 209-220, 2009.
[10] C. Carpineto, S. Osiński, G. Romano, and D. Weiss, "A Survey of Web Clustering Engines," *ACM Computing Surveys,* vol. 41, no. 3, pp. 1-38, 2009.
[11] A. Topchy, A.K. Jain, and W. Punch, "Combining Multiple Weak Clusterings," *Proc. IEEE Third Int'l Conf. Data Mining,* pp. 331-338, 2003.
[12] R. Caruana, M. Elhawary, N. Nguyen, and C. Smith, "Meta Clustering," *Proc. Sixth Int'l Conf. Data Mining,* pp. 107-118, 2006.
[13] W.M. Rand, "Objective Criteria for the Evaluation of Clustering Methods," *J. Am. Statistical Assoc.,* vol. 66, pp. 846-850, 1971.
[14] L. Hubert and P. Arabie, "Comparing Partitions," *J. Classification,* vol. 2, no. 1, pp. 193-218, 1985.
[15] N.X. Vinh, J. Epps, and J. Bailey, "Information Theoretic Measures for Clustering Comparison: Is a Correction for Chance Necessary?" *Proc. 26th Ann. Int'l Conf. Machine Learning,* pp. 1073-1080, 2009.
[16] R.J.G.B. Campello, "A Fuzzy Extension of the Rand Index and Other Related Indexes for Clustering and Classification Assessment," *Pattern Recognition Letters,* vol. 28, no. 7, pp. 833-841, 2007.
[17] A. Ben-Hur, A. Elisseeff, and I. Guyon, "A Stability Based Method for Discovering Structure in Clustered Data," *Proc. Pacific Symp. Biocomputing,* pp. 6-17, 2002.
[18] J. Barthélemy and B. Leclerc, "The Median Procedure for Partitions," *Partitioning Data Sets,* I.J. Cox, P. Hansen, and B. Julesz, eds., pp. 3-34, Am. Math. Soc., 1995.
[19] Y. Wakabayashi, "The Complexity of Computing Medians of Relations," *Resenhas,* vol. 3, no. 3, pp. 323-349, 1998.
[20] V. Filkov and S. Skiena, "Integrating Microarray Data by Consensus Clustering," *Proc. IEEE 15th Int'l Conf. Tools with Artificial Intelligence,* pp. 418-426, 2003.
[21] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering Aggregation," *ACM Trans. Knowledge Discovery from Data,* vol. 1, no. 4, 2007.
[22] T. Li and C. Ding, "Weighted Consensus Clustering," *Proc. SIAM Int'l Conf. Data Mining,* pp. 789-8092, 2008.
[23] R. Unnikrishnan and M. Hebert, "Measures of Similarity," *Proc. IEEE Seventh Workshop Applications of Computer Vision,* pp. 394-400, 2005.
[24] M. Mela, "Comparing Clusterings: An Axiomatic View," *Proc. 22nd Int'l Conf. Machine Learning,* pp. 577-584, 2005.
[25] P. Wang, C. Domeniconi, and K.B. Laskey, "Nonparametric Bayesian Clustering Ensembles," *Proc. European Conf. Machine Learning and Knowledge Discovery in Databases,* pp. 435-450, 2010.
[26] R. Bekkerman, M. Scholz, and K. Viswanatah, "Improving Clustering Stability with Combinatorial MRFs," *Proc. 15th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 99-108, 2009.
[27] X.Z. Fern and W. Lin, "Cluster Ensemble Selection," *Statistical Analysis and Data Mining,* vol. 1, no. 3, pp. 128-141, 2008.
[28] J. Azimi and X. Fern, "Adaptive Cluster Ensemble Selection," *Proc. 21st Int'l Jont Conf. Artifical Intelligence,* pp. 992-997, 2009.
[29] N. Bansal, A. Blum, and S. Chawla, "Correlation Clustering," *Proc. IEEE 43rd Ann. Symp. Foundations of Computer Science,* pp. 238-250, 2002.
[30] P. Bonizzoni, G.D. Vedova, R. Dondi, and T. Jiang, "On the Approximation of Correlation Clustering and Consensus Clustering," *J. Computer and System Sciences,* vol. 74, no. 5, pp. 671-696, 2008.
[31] C. Carpineto and G. Romano, "Optimal Meta Search Results Clustering," *Proc. 33rd Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 170-177, 2010.
[32] G.L. Liu, *Introduction to Combinatorial Mathematics.* McGraw Hill, 1968.
[33] D. Cristofor and D. Simovici, "Finding Median Partitions Using Information-Theoretical-Based Genetic Algorithms," *J. Universal Computer Science,* vol. 8, no. 2, pp. 153-172, 2002.
[34] F. Leisch, "Bagged Clustering," *Adaptive Information Systems and Modelling in Economics and Management Science,* technical report Working Papers SFB WU Vienna Univ. of Economics and Business, vol. 51, 1999.
[35] E. Dimitriadou, A. Weingessel, and F. Hornik, "A Combination Scheme for Fuzzy Clustering," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 16, no. 7, pp. 901-912, 2002.
[36] S. Osiński and D. Weiss, "A Concept-Driven Algorithm for Clustering Search Results," *IEEE Intelligent Systems,* vol. 20, no. 3, pp. 48-54, May 2005.
[37] A. Bernardini, C. Carpineto, and M. D'Amico, "Full-Subtopic Retrieval with Keyphrase-Based Search Results Clustering," *Proc. IEEE/WIC/ACM Int'l Joint Conf. Web Intelligence and Intelligent Agent Technology,* pp. 206-213, 2009.
[38] U. Manber and G. Myers, "Suffix Arrays: A New Method for On-Line String Searches," *SIAM J. Computing,* vol. 22, no. 5, pp. 935-948, 1993.
[39] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and T.K. Harshman, "Indexing by Latent Semanic Analysis," *J. Am. Soc. for Information Science,* vol. 41, no. 6, pp. 391-407, 1990.
[40] E. Ukkonen, "On-Line Construction of Suffix Trees," *Algorithmica,* vol. 14, no. 3, pp. 249-260, 1995.
[41] C. Carpineto, S. Mizzaro, G. Romano, and M. Snidero, "Mobile Information Retrieval with Search Results Clustering: Prototypes and Evaluations," *J. Am. Soc. for Information Science and Technology,* vol. 60, no. 5, pp. 877-895, 2009.
[42] K. van Rijsbergen, *Information Retrieval.* Butterworth-Heinemann, 1979.
[43] C.D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval.* Cambridge Univ. Press, 2008.

Claudio Carpineto is the head of the Information Mining Group at the Fondazione Ugo Bordoni in Rome, where he has worked since 1986. He was a visiting associate at George Mason University and at the University of Aberdeen. His current research interests include personalized and contextual information retrieval, text and pattern mining, and web knowledge extraction. He has published more than 70 papers on information retrieval, artificial intelligence, and data mining, including the book *Concept Data Analysis: Theory and Applications* (John Wiley and Sons, 2004). He is a member of the IEEE.

Giovanni Romano received the laurea degree in electronic engineering from the University of Rome "La Sapienza" From 1985 to 1986, he worked with the Institute of Psycology of National Research Council (CNR) studying and developing planning methods in the framework of human-computer interaction. Since 1986 he has been a researcher at the Fondazione Ugo Bordoni in Rome, where he has worked on the application of artificial intelligence techniques to information retrieval and data mining. He is currently involved in the development of software tools for search engines, web-based information extraction, and database access. He is a coauthor of the book *Concept Data Analysis: Theory and Applications* (John Wiley and Sons, 2004).