Pattern Recognition 43 (2010) 1943–1953

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

On voting-based consensus of cluster ensembles

Hanan G. Ayad *, Mohamed S. Kamel

Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

ARTICLE INFO

Article history:
Received 31 March 2009
Received in revised form
20 October 2009
Accepted 12 November 2009

Keywords:
Clustering
Cluster ensembles
Voting-based consensus

ABSTRACT

Voting-based consensus clustering refers to a distinct class of consensus methods in which the cluster label mismatch problem is explicitly addressed. The voting problem is defined as the problem of finding the optimal relabeling of a given partition with respect to a reference partition. It is commonly formulated as a weighted bipartite matching problem. In this paper, we present a more general formulation of the voting problem as a regression problem with multiple-response and multiple-input variables. We show that a recently introduced cumulative voting scheme is a special case corresponding to a linear regression method. We use a randomized ensemble generation technique, where an overproduced number of clusters is randomly selected for each ensemble partition. We apply an information theoretic algorithm for extracting the consensus clustering from the aggregated ensemble representation and for estimating the number of clusters. We apply it in conjunction with bipartite matching and cumulative voting. We present empirical evidence showing substantial improvements in clustering accuracy, stability, and estimation of the true number of clusters based on cumulative voting. The improvements are achieved in comparison to consensus algorithms based on bipartite matching, which perform very poorly with the chosen ensemble generation technique, and also to other recent consensus algorithms.

1. Introduction

Data clustering is a fundamental problem in exploratory data analysis, where the basic idea is to identify natural groups for unlabeled data objects. The problem has been studied for several decades in the areas of pattern recognition, machine learning, applied statistics, communications and information theory [1–9]. It arises in numerous fields of applications including data mining, text mining, bio-informatics, image analysis and segmentation, data compression, and data classification.

Data clustering is known for its inherent difficulty as an unsupervised learning problem. Over the past few years, there has been an interest in addressing the clustering problem using consensus clustering methods. Recent work on cluster ensemble methods is primarily motivated by the developments in the area of combining multiple classifiers, where the idea of voting can be readily applied. Unlike the case of classifier ensembles, the consensus of data partitions is a challenging problem due to the lack of globally defined cluster labels. Moreover, the problem of determining the optimal number of clusters is known to be generally difficult.

The goal of reconciling an ensemble of clustering solutions is to find a consensus partition that optimally summarizes the ensemble and to obtain a clustering solution with improved accuracy and stability compared to the individual members of the ensemble. Naturally, the quality of a consensus clustering highly depends on the ensemble generation technique and on the effectiveness of the consensus method in combining the generated ensemble.

Generally, a consensus clustering method (a.k.a. consensus function) constructs an aggregate representation of the ensemble and use it as the basis for extracting a consensus partition. Ensemble representations are usually designed so as to circumvent the cluster label correspondence problem [12–14]. Examples of such representations are the objects' pairwise co-association or co-occurrence matrix [13]; hyper-graphs and meta-graphs [12]; and categorical feature-spaces [14]. On the other hand, voting-based methods attempt to establish direct parallels with consensus methods for multiple classifiers, by seeking an optimal relabeling of the ensemble partitions [15–20]. The idea of relabeling is to match the symbolic cluster labels between the different ensemble partitions. The relabeling enables the aggregation via averaging of the objects' cluster-label assignments into an ensemble representation consisting of a central (or a median) aggregated partition.

Viewed from the relabeling perspective, voting-based methods represent a distinct class of consensus methods. The simultaneous

* Corresponding author.
E-mail address: hanan@pami.uwaterloo.ca (H.G. Ayad).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.11.012

relabeling and aggregation of an ensemble is a computationally difficult problem [15,21] and is generally addressed via pairwise relabeling of each ensemble partition with respect to a reference partition. This pairwise relabeling is referred to here as the voting problem. In the cluster ensemble literature, the voting problem is commonly formulated as a weighted bipartite matching problem [15–19]. In this paper, we present a general formulation of the voting problem as a multi-response regression problem. We show that bipartite matching and a recent cumulative voting scheme introduced in [20] represent special cases of this more general formulation. The latter corresponds to fitting a linear model by least squares estimation and the former is a more constrained least squares problem (a.k.a. linear sum assignment problem LSAP). In particular, the paper demonstrates that the effectiveness of these different special cases highly depends on the ensemble generation technique.

The voting-based aggregated representation is a soft partition. In general, the term "soft" is used in the literature either to describe a partition obtained using a statistical model-based clustering algorithm that maximizes the likelihood function [22,23], where u_{jq} reflects the uncertainty about the associated classification of each data object or to describe a partition obtained using a clustering algorithm that optimizes a fuzzy objective function [24,25], where u_{jq} reflects a fuzzy membership value. In this paper, the aggregated partition is viewed as a soft partition in a statistical sense. Specifically, it is obtained via averaging of cluster-label assignment probabilities.

We use the ensemble generation techniques proposed in [13,26], where the number of clusters for each partition may vary and is generally greater than the number of true or desired clusters. Given this technique, the aggregated ensemble partition obtained by relabeling and averaging of cluster-label assignments is viewed as a distributional (statistical) representation of the ensemble. This view leads to formulating the problem of extracting an optimal consensus clustering as that of finding an optimal compression of this statistical distribution such that maximum amount of information is preserved. The information-bottleneck based algorithm described in [20] is applied here to extract the consensus clustering, where each of the bipartite matching and cumulative voting schemes are considered. Experimental results demonstrate substantially more accurate consensus solutions and better estimates of the number of clusters in the case of cumulative voting, when used in conjunction with the employed ensemble generation technique.

In principle, the voting schemes described here can be applied to hard or soft ensembles. The bipartite matching scheme has been applied to both hard [17–19] and soft ensembles [15,16], and basic modifications required for applying cumulative voting to soft ensembles were briefly described in [20]. However, for simplicity and a focused analysis, we assume hard ensembles as input. Analysis based on soft ensembles is beyond the scope of this paper.

2. The voting problem

2.1. Notations and overview

Let $\mathcal{X}=\{x_j\}_{j=1}^{n}$ denote a set of n data objects where each is a vector $\mathbf{x}\in\Re^d$. A clustering algorithm takes as input a data matrix $\mathbf{X}$ and partitions $\mathcal{X}$ into k clusters, whereby each $x\in\mathcal{X}$ is assigned a cluster label in $\mathcal{C}=\{c_q\}_{q=1}^{k}$. A partition may be represented by an n-dimensional labeling vector $\mathbf{y}\in\mathcal{C}^n$ or an $n\times k$ stochastic matrix $\mathbf{U}$, with a row for each object, and a column for each cluster, such that $\sum_{q=1}^{k}u_{jq}=1,\forall j$. In general, $\mathbf{U}$ may represent a hard partition with $u_{jq}\in\{0,1\}$ or a soft partition with $u_{jq}\in[0,1]$.

Let $\mathcal{U}=\{\mathbf{U}^i\}_{i=1}^{b}$ denote an ensemble of b partitions, where each partition consists of a number of clusters k_i. The voting-based aggregation problem can be expressed as an optimization problem consisting of two sub-problems [15], as given by Eq. (1). The first sub-problem is to seek the most consistently relabeled partitions, denoted by $\{\vartheta(\mathbf{U}^i)\}_{i=1}^{b}$, in the sense of being as close as possible to each others according to a defined distance function d. The second sub-problem is to find a soft aggregated-partition $\overline{\mathbf{U}}$ that is most agreeing or closest to the relabeled ensemble partitions (i.e., $\overline{\mathbf{U}}$ should minimize the sum of the distances from $\{\vartheta(\mathbf{U}^i)\}_{i=1}^{b}$). The first sub-problem is referred to here as the relabeling problem and the second as the aggregation problem:

$$\min_{\overline{\mathbf{U}}}\min_{\vartheta(\mathbf{U}^i)}\sum_{i=1}^{b}d(\overline{\mathbf{U}},\vartheta(\mathbf{U}^i)). \tag{1}$$

As noted in [15], the problem in Eq. (1) requires the simultaneous optimization of $\vartheta^i(\mathbf{U}^i)$ with respect to $\overline{\mathbf{U}}$ and of $\overline{\mathbf{U}}$ with respect to $\vartheta^i(\mathbf{U}^i)$. Assuming fixed $\vartheta^i(\mathbf{U}^i)$, the optimal $\overline{\mathbf{U}}$ is the soft partition computed as the average $(1/b)\sum_{i=1}^{b}\vartheta^i(\mathbf{U}^i)$. Hornik [21] notes that finding the optimally permuted partitions, in a global sense, corresponds to a multi-dimensional assignment problem (MAP), which unlike LSAP, is NP-hard, with branch-and-bound approaches being computationally intractable for typical ensemble sizes ($b\geq 20$). An efficient iterative algorithm is derived in [15], where at each iteration the optimal relabeling of an ensemble partition is determined with respect to the last aggregated $\overline{\mathbf{U}}$, followed by an update of $\overline{\mathbf{U}}$.

In general, the voting-based aggregation problem is dealt with using voting algorithms that employ iterative pairwise relabeling [15,17,19,20]. Thus, a fundamental element of dealing with the problem is the pairwise relabeling (voting) problem. A relabeled partition is viewed as representing "votes" for the assignments of cluster labels to the data objects.

2.2. General problem formulation

Let $\mathbf{U}^0$ denote a partition designated as a reference and $\mathbf{U}^i$ denotes an arbitrary input partition. The problem is to find the optimal relabeling of $\mathbf{U}^i$, denoted $\vartheta^i(\mathbf{U}^i)$ or $\mathbf{V}^i$, that is as closest to $\mathbf{U}^0$, according to d.

The problem is commonly formulated as a weighted bipartite matching problem, which is a well-known combinatorial optimization problem that is solvable in $O(k^3)$, using the Hungarian method [27]. Many distance functions can be defined and the most common is the Euclidean distance. A relabeled partition $\vartheta(\mathbf{U}^i)$ is defined as a column permutation of $\mathbf{U}^i$. That is, $\vartheta(\mathbf{U}^i)=\mathbf{U}^i\mathbf{W}^i$ where $\mathbf{W}^i$ is a permutation matrix. The problem is to find $\mathbf{W}^i$ that minimizes the sum of squared errors given as follows:

$$\min_{\mathbf{W}^i}\quad \|\mathbf{U}^0-\mathbf{U}^i\mathbf{W}^i\|^2$$
$$\text{s.t.}\quad \sum_{l=1}^{k}w_{lq}^i=\sum_{q=1}^{k}w_{lq}^i=1\quad \text{where } w_{lq}^i=0 \text{ or } 1, \tag{2}$$

where $k=\max(k_i,k_0)$. It is noted that in the case of bipartite matching, the ensemble is often constrained to partitions with a fixed number of clusters k that is also equal to the desired number of consensus clusters as in [17–19]. Alternatively, empty (dummy) clusters need to be added to the partition with a smaller number of clusters as proposed in [15,16].

The weighted bipartite matching formulation makes intuitive sense because the relabeled partition $\vartheta(\mathbf{U}^i)$ is identical to $\mathbf{U}^i$ except that the cluster labels are permuted to optimally match with other ensemble partitions. Furthermore, theoretical arguments presented in [19] prove the optimality of the aggregated partition based on bipartite matching in conjunction with a specific ensemble generation technique described in [19]. However, we argue that the bipartite matching formulation is not suitable when used in conjunction with alternative ensemble generation techniques such as those considered in this paper. We propose a more general formulation of the voting problem and demonstrate the superiority of a special instance of this formulation for the ensemble generation technique considered in this paper.

We formulate the problem as a multiple regression problem with multiple output (response) variables and multiple input variables. The response variables $\{C_q^0\}_{q=1}^{k_0}$ are represented by the clusters of $\mathbf{U}^0$ and the input variables $\{C_l^i\}_{l=1}^{k_i}$ are represented by the clusters of partition $\mathbf{U}^i$. In other words, the problem can be generalized to a supervised learning problem with continuous response variables, which leads to a soft relabeled partition $\vartheta(\mathbf{U}^i)$.

Specifically, the voting problem is stated as that of estimating the assignments of the objects to the reference clusters $\mathcal{C}^0=\{c_1^0,\ldots,c_{k_0}^0\}$, given their assignments to the clusters of an ensemble partition $\mathcal{C}^i=\{c_1^i,\ldots,c_{k_i}^i\}$, such that the estimation errors compared to the representative partition $\mathbf{U}^0$ are minimized. Let the random vectors $C^i=(C_1^i\ldots C_{k_i}^i)$ and $C^0=(C_1^0\ldots C_{k_0}^0)$ denote the clusters of $\mathbf{U}^i$ and $\mathbf{U}^0$, respectively, where each variable C_q^0 is considered to take real values in [0,1], such that $\sum_{q=1}^{k_0} u_{jq}^0=1$. That is, $\mathbf{U}^0$ is generally considered a soft partition. The voting problem can be viewed as seeking a function of $\mathbf{U}^i$, $\vartheta^i(\mathbf{U}^i)$, that establishes a relationship between $\{C_l^i\}_{l=1}^{k_i}$ and $\{C_q^0\}_{q=1}^{k_0}$, such that a loss function $L^i(\mathbf{U}^0,\vartheta^i(\mathbf{U}^i))$ is minimized, where L^i is referred to here as the voting (or pairwise relabeling) loss. The function $L^i(\mathbf{U}^0,\vartheta^i(\mathbf{U}^i))$ penalizes the errors in the estimated values of $\vartheta^i(\mathbf{U}^i)$ compared to $\mathbf{U}^0$. The problem of finding $\vartheta^i(\mathbf{U}^i)$ is given by

$$\min_{\vartheta^i(\mathbf{U}^i)} L^i(\mathbf{U}^0,\vartheta^i(\mathbf{U}^i)).$$

Based on this formulation, the regression function $\vartheta^i(\mathbf{U}^i)$, which may be referred to as the voting (or relabeling) function, estimates the conditional expectation of C^0 given C^i, $\mathrm{E}(C^0|C^i)$, and is a vector function, $\vartheta^i(\mathbf{U}^i)=(\vartheta_1^i(\mathbf{U}^i),\ldots,\vartheta_{k_0}^i(\mathbf{U}^i))$ [28]. Let $\mathcal{L}^i$ denotes the *i*-th *learning set* corresponding to the partition pair $\mathbf{U}^i$ and $\mathbf{U}^0$, and consisting of the vectors $\{(\mathbf{u}_j^i,\mathbf{u}_j^0)\}_{j=1}^n$, where $\mathbf{u}_j^i$ is the *j* th row vector of $\mathbf{U}^i$ and represents a k_i input vector, and $\mathbf{u}_j^0$ is the *j*-th row vector of $\mathbf{U}^0$ and represents a k_0 output *target* vector. The goal is to use $\mathcal{L}^i$ to estimate $\vartheta^i(\mathbf{U}^i)$. In the voting problem, it is only the learning but not the prediction aspect of regression that is applied.

In a regression problem, the form of the function $\vartheta^i(\mathbf{U}^i)$, that underlies the relationship between the input and output variables, is generally unknown [28]. A simple but often reasonable regression method is to fit a linear model by least squares estimation. It is considered reasonable for its minimal assumptions about the underlying model of the data. As described below, the cumulative voting scheme introduced in [20] is a special instance of the regression problem that corresponds to fitting a linear model.

It is noted that in fitting regression models, an inverse problem is implicitly defined [10]. Thus, ill-posedness may arise, whereby small perturbations in the target vector imply large changes in the estimated solution. Ill-posedness may arise if the number of parameters to be estimated exceeds the number of observations [11]. Regularization techniques can be required to address this problem. In this paper, regularization has not been applied.

2.3. Cumulative voting

In [20], two types of cumulative voting are investigated; the normalized and un-normalized schemes. The term "cumulative" refers to the property that the computed vote weights for each variable C_q^0 must add up to a prespecified value. In the case of the normalized scheme, the sum must be 1, and in the case of the un-normalized scheme, the sum must be equal to the size of the voting cluster. In this paper, we focus on the normalized scheme.

Assuming a linear model for each output variable, $\vartheta^i(\mathbf{U}^i)$ is written in matrix notation as $\vartheta^i(\mathbf{U}^i)=\mathbf{U}^i\mathbf{W}^i$, where $\mathbf{W}^i$ is a $k_i\times k_0$ matrix of coefficients denoted as w_{lq}^i. Let $\mathbf{V}^i=\vartheta^i(\mathbf{U}^i)$, which is an $n\times k_0$ matrix. To fit the linear model to the learning set $\mathcal{L}^i$, the coefficients $\mathbf{W}^i$ are estimated to minimize the mean squared errors, as given in

$$L^i(\mathbf{U}^0,\vartheta^i(\mathbf{U}^i))=\mathrm{MSE}^i(\mathbf{U}^0,\vartheta^i(\mathbf{U}^i))=\frac{1}{n}\sum_{j=1}^{n}\sum_{q=1}^{k_0}(u_{jq}^0-v_{jq}^i)^2, \tag{3}$$

which is written in matrix notation as follows:

$$\mathrm{MSE}^i(\mathbf{U}^0,\vartheta^i(\mathbf{U}^i))=\frac{1}{n}\mathrm{tr}[(\mathbf{U}^0-\mathbf{U}^i\mathbf{W}^i)^T(\mathbf{U}^0-\mathbf{U}^i\mathbf{W}^i)]. \tag{4}$$

The solution is obtained by differentiating with respect to $\mathbf{W}^i$ and is given by Eq. (5). The estimated partition $\hat{\mathbf{V}}^i$ is given by $\hat{\mathbf{V}}^i=\mathbf{U}^i\hat{\mathbf{W}}^i$:

$$\hat{\mathbf{W}}^i=(\mathbf{U}^{iT}\mathbf{U}^i)^{-1}\mathbf{U}^{iT}\mathbf{U}^0. \tag{5}$$

It is easy to see that the normalized scheme in [20] corresponds to the linear model with least squares fit, as defined above by noting that, for hard ensemble partitions, the term $(\mathbf{U}^{iT}\mathbf{U}^i)$ in Eq. (5) is a diagonal matrix, and hence Eq. (5) gives the same expression for the coefficients as computed in [20], which is given by, $\hat{w}_{lq}^i=(1/n_{l.}^i)\sum_{j\in\{1,\ldots,n\}:u_{jl}^i=1}u_{jq}^0$, where $n_{l.}^i$ denotes the number of objects assigned to cluster c_l^i, and $\hat{v}_{jq}^i=\hat{w}_{lq}^i$ if $u_{jl}^i=1$, and 0 otherwise. If $\mathbf{U}^0$ is also a hard partition, then $\hat{w}_{lq}^i=n_{lq}^i/n_{l.}^i$, where n_{lq}^i is the number of objects assigned to clusters c_l^i and c_q. Note that $\mathbf{U}^0$ a hard reference partition if the aggregation algorithm uses fixed-reference, whereby an initially selected partition is used as a common reference for all the ensemble partitions and it remains unchanged throughout the aggregation procedure. However, in this paper, we apply a stepwise algorithm.

Based on the constraint on the output variables $\{C_q^0\}_{q=1}^{k_0}$, the estimated values $\hat{v}_{jq}^i$ must sum to 1, $\sum_{q=1}^{k_0}\hat{v}_{jq}^i=1$. Note that one may consider C^0 as a categorical variable with k_0 categories, $\{c_1^0,\ldots,c_{k_0}^0\}$. Based on the squared error loss estimation, the estimate $\hat{\mathbf{V}}^i$ of the conditional expectation can be viewed as an estimate of the posterior probability $\mathrm{E}(C^0|C^i)=\Pr(C^0|C^i)$ [28]. Classifying to the most probable class $\hat{\mathbf{V}}^i=\arg\max_{c_q^0\in C^0}\Pr(c_q^0|C^i)$ gives the *Bayes classifier*, with the *Bayes rate* as the error rate [28]. Hence, this rate gives a lower bound on the achievable error rate for a relabeling $\vartheta^i(\mathbf{U}^i)$, based on least squares loss. The error rate is denoted as $\mathrm{Err}^i(\mathbf{U}^0,\vartheta^i(\mathbf{U}^i))$. It corresponds to the case where C^0 is

considered a categorical variable, i.e., when the voting problem is viewed as a classification problem.

For the bipartite matching formulation, the minimization problem in Eq. (2) is equivalent to the constrained maximization of $\mathrm{tr}(\mathbf{G}^{iT}\mathbf{W}^i)$ [15], where $\mathbf{G}^i$ is the contingency matrix of $\mathbf{U}^i$ and $\mathbf{U}^0$, $\mathbf{G}^i = \mathbf{U}^{iT}\mathbf{U}^0$. Unlike cumulative voting, if $\mathbf{U}^i$ is a hard partition, the computed $\mathbf{V}^i$ is also hard. In the case of hard ensemble partitions $\{\mathbf{U}^i\}_{i=1}^b$, the problem is equivalent to minimizing the probability of error p_e^i subject to the constraints defined in Eq. (2), where p_e^i is given by $p_e^i = (1/n)\sum_{l=1}^k \sum_{q=1}^k g_{lq}^i(1-w_{lq}^i)$.

If both $\mathbf{U}^i$ and $\mathbf{U}^0$ are hard partitions, the constrained error rate denoted as $\mathrm{Err}^i(\mathbf{U}^0, \vartheta^i(\mathbf{U}^i))$ is minimized, which is given by $(1/n)\sum_{l=1}^k \sum_{q=1}^k n_{lq}^i(1-w_{lq}^i)$, where n_{lq}^i is the number of objects assigned to clusters c_l^i and c_q^0.

It is easy to see that due to the additional constraints in the case of bipartite matching, the error rate achievable based on bipartite matching is bounded from below by the error rate achievable using cumulative voting when the latter is followed by classifying to the most probable class.

Bipartite matching establishes binary one-to-one relations between two sets of clusters, whereas the new formulation considers the idea of a soft relabeling (or soft voting) and establishes real-valued many-to-many relations between the clusters of a given partition and those of a reference partition. The generality of the proposed formulation enables the modeling of more complex relations arising in cases of substantial variability among the ensemble partitions, one of which is a variable number of clusters. The cumulative voting scheme is used here as the vehicle for demonstrating the validity of this idea. Below is an illustrative example of the different voting schemes.

2.4. Illustrative example

Consider a set of 10 data objects $\mathcal{X} = \{x_1, \ldots, x_{10}\}$. Suppose a reference partition $\mathbf{U}^0$ and an ensemble partition $\mathbf{U}^i$ are given as follows, where $\mathbf{U}^0$ and $\mathbf{U}^i$ partition the objects into $k_0 = 5$ and $k_i = 2$ clusters, respectively. Relabeling $\mathbf{U}^i$ based on cumulative voting gives a coefficient matrix $\mathbf{W}^i$ and a relabeled partition $\mathbf{V}^i$ as shown below.

That is, in the case of cumulative voting, the uncertainties associated with assigning the objects belonging to cluster 1 of $\mathbf{U}^i$ to each of the five clusters of $\mathbf{U}^0$ are given by first four rows of $\mathbf{V}^i$. This soft assignment reflects the fact that the objects belonging to cluster 1 of $\mathbf{U}^i$ are divided equally among the first two clusters of $\mathbf{U}^0$. Similarly, equal probabilities of assigning the objects belonging to cluster 2 of $\mathbf{U}^i$ to each of the last three reference clusters are reflected in the last six rows of $\mathbf{V}^i$:

$$\mathbf{U}^0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{U}^i = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

$$\mathbf{W}^i = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.3333 & 0.3333 & 0.3333 \end{bmatrix} \quad \text{and}$$

$$\mathbf{V}^i = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.3333 & 0.3333 & 0.3333 \\ 0 & 0 & 0.3333 & 0.3333 & 0.3333 \\ 0 & 0 & 0.3333 & 0.3333 & 0.3333 \\ 0 & 0 & 0.3333 & 0.3333 & 0.3333 \\ 0 & 0 & 0.3333 & 0.3333 & 0.3333 \\ 0 & 0 & 0.3333 & 0.3333 & 0.3333 \end{bmatrix}.$$

Based on bipartite matching, relabeling $\mathbf{U}^i$ gives a permutation matrix $\mathbf{W}^i$ and a relabeled partition $\mathbf{V}^i$ as follows:

$$\mathbf{W}^i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{V}^i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

For the bipartite matching scheme, the objects belonging to cluster 1 of $\mathbf{U}^i$ are simply assigned to reference clusters 1, and no objects are assigned to reference cluster 2. Similarly the objects of cluster 2 of $\mathbf{U}^i$ are assigned to reference clusters 3, and no objects are assigned to cluster 4 or 5. The relabeling requires three empty clusters to be created. Unlike the cumulative voting scheme, the assignment of the data objects ignores the fact that each cluster of $\mathbf{U}^i$ is equally divided between more than one reference cluster.

3. Aggregation and extraction of consensus clustering

3.1. Aggregation

Using the least squares objective for the aggregated partition, with respect to the ensemble partitions, the aggregation problem is written as

$$\min_{\overline{\mathbf{U}}} \mathrm{MSE}(\overline{\mathbf{U}}; \mathbf{U}^1, \ldots, \mathbf{U}^b) = \min_{\overline{\mathbf{U}}} \min_{\vartheta^i(\mathbf{U}^i)} \frac{1}{b}\sum_{i=1}^{b} \mathrm{MSE}^i(\overline{\mathbf{U}}, \vartheta^i(\mathbf{U}^i)). \tag{6}$$

Several voting-based aggregation algorithms, which are computationally efficient, are described in recent literature [15–20]. In the simplest approach, one common reference partition $\mathbf{U}^0$ is selected and each of $\{\mathbf{U}^i\}_{i=1}^b$ is optimally re-labeled with respect to $\mathbf{U}^0$. Then, $\overline{\mathbf{U}}$ is computed by averaging the $\{\mathbf{V}^i\}_{i=1}^b$. As observed in [18], the drawback of this algorithm is its high dependency on the selected (fixed) reference $\mathbf{U}^0$. However, it is noted that this algorithm represents a suitable approach if the ensemble partitions are known to be uniform. For instance, in the case of the stochastic partition generation model described in [19], where all the ensemble partitions are generated as noisy permutations of an underlying clustering, according to a probability of error, a proof of the convergence of the aggregated partition to the underlying (presumably true) clustering is presented in [19].

Dimitriadou et al. [15] present an iterative algorithm to find an approximate solution for the aggregation problem in Eq. (6). The algorithm works as follows. An initial reference is set as $\mathbf{U}^0 = \mathbf{U}^1$. Then, at each step i, for $i \in \{2, \ldots, b\}$, the locally optimal re-labeling $\vartheta^i(\mathbf{U}^i)$ with respect to the current reference partition is computed,

and $\mathbf{U}^0$ is re-computed as the weighted average of the last $\mathbf{U}^0$ and $\vartheta^i(\mathbf{U}^i)$, such that the re-computed reference represents the average of the partitions relabeled thus far (at step i/b). Similar greedy approximation algorithms were also described in [18,?]. It is noted that for this algorithm, the obtained solution $\overline{\mathbf{U}}$ depends on the ordering of the partitions, and the initial reference $\mathbf{U}^0$. The algorithm can be enhanced by running several passes with random initialization and random order of the partitions, and keeping the best solution [21].

In this paper, the enhanced iterative algorithm described in Algorithm 1 is applied in conjunction with the bipartite matching scheme and is referred to as `bVote`. Several passes can be performed by running `bVote` multiple times with random ordering of the ensemble partitions, and keeping the best solution achieved so far (i.e. $\overline{\mathbf{U}}$ with lowest value of MSE($\overline{\mathbf{U}},\mathcal{U}$) is kept).

Algorithm 1. `bVote`.

Function $\overline{\mathbf{U}} = \texttt{bVote}(\mathcal{U})$
1: Randomly select a partition $\mathbf{U}^i \in \mathcal{U}$ and assign to $\mathbf{U}^0$
2: **for** $i = 1$ to b **do**
3: Compute $\mathbf{W}^i$ by finding the bipartite matching solution to Eq. (2).
4: $\mathbf{V}^i = \mathbf{U}^i\mathbf{W}^i$
5: $\mathbf{U}^0 = \frac{i-1}{i}\mathbf{U}^0 + \frac{1}{i}\mathbf{V}^i$
6: **end for**
7: $\overline{\mathbf{U}} = \mathbf{U}^0$.

For the cumulative voting scheme, we apply the algorithm described in [20], and which we further motivate here. We refer to it as adaptive cumulative voting, `Ada-cVote`, and outline it in Algorithm 2. The aggregated distribution represented by $\overline{\mathbf{U}}$ correspond to the conditional probability distribution $\overline{p}(c^0|x)$. To simplify the notation, we refer to the distribution $\overline{p}(c^0|x)$ as $p(c|x)$. In other words, the random variable C^0 is referred to here as C.

Let $H(C)$ denote the Shannon entropy associated with C, which is sometimes written as $H(p(c))$. Defined over the cluster labels of the partition $\overline{\mathbf{U}}$, $H(C)$ measures the average amount of information associated with C and is defined as a function of its distribution $p(c)$ as follows [29], $H(C) = -\sum_{c\in C} p(c)\log p(c)$. Let $I(C;X)$ denote the mutual information between C and X. $I(C;X)$ measures the amount of information that the random variable C contains about X, and vice versa. It is defined as

$$I(C;X) = \sum_c \sum_x p(c,x)\log\frac{p(c,x)}{p(c)p(x)}, \tag{7}$$

and can also be written as $I(C;X) = H(C) - H(C|X)$.

It is noted that for a hard partition $\mathbf{U}^i$, we have $I(C^i;X) = H(C^i)$, since the value of C^i is completely determined by the value of X (i.e., $H(C^i|X) = 0$). It follows that $I(C;X)$ is bounded from above by $H(C)$; $I(C;X) \le H(C)$, where $H(C) = H(C^0)$. Thus, the initially selected reference partition determines the following measures: the entropy associated with the aggregated clusters, the initial value of the mutual information $I(C^0;X)$, and the upper bound on the amount of information that random variable C contains about X. This result motivates the use of a selection criterion for the initial reference partition based on the mutual information $I(C^i;X)$, which is equal to $H(C^i)$ for hard partitions, as given by

$$\mathbf{U}^0 = \underset{\mathbf{U}^i\in\mathcal{U}}{\operatorname{argmax}}\, I(C^i;X) \equiv \underset{\mathbf{U}^i\in\mathcal{U}}{\operatorname{argmax}}\, H(C^i). \tag{8}$$

The aggregation can also be further improved if the algorithm greedily selects at each aggregation step i the ensemble partition that keeps the mutual information $I_i^0(C^0;X)$ as close as possible to $I_{i-1}^0(C^0;X)$, where $I_i^0(C^0;X)$ and $I_{i-1}^0(C^0;X)$ are associated with the reference partitions computed at step i and step $i-1$, and representing the distributions denoted here by $p_i^0(c^0|x)$ and $p_{i-1}^0(c^0|x)$, respectively. This greedy aggregation sequence limits the loss in $I(C;X)$ for the aggregated partition.

Algorithm 2. `Ada-cVote`.

Function $\overline{\mathbf{U}} = \texttt{Ada}-\texttt{cVote}(\mathcal{U})$
1: Re-order $\mathbf{U}^i \in \mathcal{U}$ in decreasing order of $I(C^i;X)$ ($\equiv H(C^i)$ for hard partitions)
2: Assign $\mathbf{U}^1$ to $\mathbf{U}^0$.
3: **for** $i = 2$ to b **do**
4: Compute $\mathbf{W}^i$ as given by Eq. (5).
5: $\mathbf{V}^i = \mathbf{U}^i\mathbf{W}^i$
6: $\mathbf{U}^0 = \frac{i-1}{i}\mathbf{U}^0 + \frac{1}{i}\mathbf{V}^i$
7: **end for**
8: $\overline{\mathbf{U}} = \mathbf{U}^0$.

Note that $p_i^0(c^0,x)$ is the average of the relabeled partitions up to the i-th iteration, as described in Algorithm 2, which is given as follows:

$$p_i^0(c^0|x) = \frac{i-1}{i}p_{i-1}^0(c^0|x) + \frac{1}{i}p^i(c^0|x). \tag{9}$$

The mutual information $I_i^0(C^0;X)$ is also written in terms of the *Kullback–Leibler* divergence [29] $D(\cdot\|\cdot)$, (a.k.a. *relative entropy*), between the joint $p_i^0(c^0,x)$ and the product distribution $p(c^0)p(x)$ as: $D(p_i^0(c^0,x)\|p(c^0)p(x))$. Since $p(c^0)$ and $p(x)$ remain constant for the cumulative voting scheme, the goal is to select $\mathbf{U}^i$ that leads to $p^i(c^0|x)$ being as close as possible to the weighted $p_{i-1}^0(c^0|x)$. That is, $\mathbf{U}^i$ should be selected such that the divergence between $p^i(c^0|x)$ and $p_{i-1}^0(c^0|x)$ is minimized. Note that the averaging formula in `Ada-cVote` is slightly adjusted from [20] to match with `bVote` [15].

The `Ada-cVote` is an efficient heuristic algorithm that seeks to minimize the divergence criterion by choosing at each aggregation step i, the ensemble partition $\mathbf{U}^i$ that maximizes $I(C^i;X)$, or equivalently $H(C^i)$, for hard ensembles. It saves computational time since the entropies can be computed once for each partition, prior to aggregating, rather than computing at each step i, the divergences between the current reference and all the remaining partitions, after relabeling. Furthermore, the simplified criterion represents a reasonable heuristic given the ensemble generation mechanism that is considered in this paper. As described in Section 4, the same base algorithm (k-means) is applied to generate the ensemble partitions, with a randomly selected number of clusters. Therefore, we assume that the closer the values of $H(C^i)$, the more similar the cluster structures of corresponding $\mathbf{U}^i$, and hence, one can obtain the least amount of information loss, or equivalently, minimum divergence from the current reference distribution is obtained.

A good feature of the `Ada-cVote` algorithm is that the resulting aggregated partition is invariant of the order of the input partitions and of the initially selected partition. This invariability is a desirable property for an aggregation algorithm and it also saves the extra computations required to enhance an algorithm such as `bVote`, where multiple passes may be needed.

3.2. Extraction of consensus clustering

Rather than considering $\overline{\mathbf{U}}$ itself as the consensus partition, it is viewed as an optimized ensemble representation. Then, the goal is to extract a coherent and global cluster structure $\hat{\mathbf{U}}$ for the data based on this distributional representation given by $\overline{\mathbf{U}}$. We apply the efficient O(n) agglomerative algorithm proposed in [20] based

Table 1
Characteristics of the datasets and ARI values for the k-means (mean $\pm$ std).

Dataset	n	d	k	Class distribution	k-Means ARI
Artificial datasets					
2D2K	1000	2	2	50% each	0.92 ± 0.00
8D5K	1000	8	5	20% each	0.86 ± 0.16
Two Gauss	300	2	2	33, 67%	0.81 ± 0.00
Real datasets					
Yahoo!	2340	1458	6	$\approx 6, 59, 21, 5, 6, 3\%$	0.42 ± 0.08
Landsat (Statlog)	6435	36	6	$\approx 24, 11, 22, 9, 11, 23\%$	0.46 ± 0.08
E.coli proteins	336	7	8	$\approx 43, 23, 15.4, 10, 6, 1.4, 0.6, 0.6\%$	0.39 ± 0.04

on the information bottleneck method in [8,30]. The optimal partition $\hat{\mathbf{U}}$ is defined as the most compressed summary of $\overline{\mathbf{U}}$ that preserves maximum amount of relevant information about the data. The algorithm, which is detailed in [20], minimizes the average Jensen–Shannon divergence within the cluster and produces a hierarchy of k-partitions, for $k \in \{\overline{k}, \overline{k}-1, \ldots, 1\}$. We refer to it as `JS-Alink`. It is applied to obtain $\hat{\mathbf{U}}$, for either an estimated or a predetermined number of clusters $\hat{k} \leq \overline{k}$.

If $\hat{k}$ is predetermined, the $\hat{k}$- partition is obtained from the dendrogram and computed as detailed in [20]. When looking for an optimal $\hat{k}$ value, we apply the idea of a k-cluster *lifetime* described in [13] on the merging JS divergence values. Specifically, the optimal $\hat{k}$ is defined as the number of clusters with the longest lifetime, where the lifetime of each k is defined as the range of distance threshold values that lead to a k-partition. It is computed as the difference between the minimum and maximum distances that lead to merging the input patterns into k clusters. In other words, the life time of k is the difference between the merging distance leading to a k-partition and that leading to a $(k-1)$- partition in the obtained dendrogram.

Note that in the case where $\overline{k}$ may be smaller than $\hat{k}$, the agglomerative information bottleneck algorithm, described in [30], can be applied. The algorithm generates a hierarchy of partitions for n to 1 clusters. Unlike the algorithm in [20], the computational complexity in this case is $O(n^2)$. In this paper, we do not consider this case because the co-association based consensus algorithms in [13] represent a simpler $O(n^2)$ alternative, whereas we seek here to provide more efficient $O(n)$ algorithms.

4. Empirical study

4.1. Consensus algorithms

The focus of the empirical study is on comparing voting-based consensus algorithms derived from the objective functions presented in Section 2, namely, bipartite matching and the instance of cumulative voting that corresponds to fitting a linear regression model. Each of the `Ada-cVote` and `bVote` algorithms is applied in conjunction with the agglomerative algorithm `JS-Alink`. In both cases, the algorithms are evaluated for extracting a consensus partition with a predetermined k, in which case they are abbreviated as ACV-k and BV-k, for the adaptive cumulative voting and bipartite matching schemes, respectively. They are also compared for extracting a consensus partition with an estimated k, using k-cluster lifetimes, where they are abbreviated as ACV and BV, respectively. Comparison with other recent consensus algorithms is also presented, for further validation and help to provide a bigger picture of the algorithm's performances.

All algorithms have been implemented using MATLAB. The evidence accumulation consensus algorithms (EAC) [13], with hierarchical single link and average link algorithms are applied, where the corresponding algorithms are referred to as EAC-S and EAC-A, respectively. For the EAC algorithms, the co-association matrix is computed and a distance function (1-co-association ratio) is calculated and used as input for the hierarchical algorithms. For the graph-based algorithms: CSPA, HGPA, and MCLA [12], the implementation provided at the authors' website[1] was used. For the quadratic mutual information algorithm, QMI [14,31], it was implemented as specified by the authors. A standardization is applied to transform the cluster labels into quantitative features by replacing the i-th partition by k_i binary features. Then, each binary feature is standardized to a zero mean. The k-means algorithm is applied on the transformed data to find the consensus clustering (10 runs for the k-means algorithm are performed and the clustering with minimum mean squared error is selected).

4.2. Datasets

Table 1 summarizes the characteristics of the datasets used in the experiments, along with the accuracy as measured by the adjusted Rand Index [32] for the k-means algorithm (or spherical k-means for text data) compared to the true clustering, where k is set to the true number of clusters.

The first two artificial datasets were generated and used in previous related work [12,26]. We generated the third as a mixture of two Gaussian clusters with unbalanced sizes, different means and covariance matrices, and a slight overlap. The real datasets are: the Yahoo! data,[2] the Lansat data and the *Escherichia coli* proteins [34], where the last two are available from the UCI machine learning repository [35]. The Yahoo! dataset consists of documents parsed from news web-pages and represented as term frequency matrix. A human classification of the documents into a six primary categories (Business, Entertainment, Health, Politics, Sports, and Technology). The Landsat dataset was generated by taking a section from data purchased from NASA by the Australian Centre for Remote Sensing, and used for research at The Centre for Remote Sensing University of New South Wales [35]. The dataset consists of the multi-spectral values of pixels in 3×3 neighbourhoods in a satellite image, and the classification of the central pixel in each neighbourhood into one of six classes of soil types. The *E. coli* proteins data were created at the National Institute of

[1] http://www.strehl.com/
[2] Data available at http://ftp.cs.umn.edu/dept/users/boley/PDDPdata/doc-K/

Molecular and Cellular Biology in Japan; proteins from *E. coli* are classified into eight classes according to cellular localization sites, based on seven features calculated from the amino acid sequences.

4.3. Ensemble generation technique

We apply the ensemble generation technique presented in [13,26], where an ensemble consists of partitions with overproduced clusters (i.e., k_i is larger than the desired or suspected number of clusters), and where the number of overproduced clusters may either be fixed or randomly selected for each partition. For a variable number of clusters, k_i is randomly selected in a range $[k_{\min}, k_{\max}]$. In general, the range $[k_{\min}, k_{\max}]$ can be varied in a search for stable consensus partitions. In the experiments, $k_{\min}$ is simply selected as a multiple of the desired number of clusters, and $k_{\max}$ is set to some relatively larger value. The technique was proposed in [26] as a means of inducing diversity among the ensemble partitions (according to several presented measures), where experiments show that it increases the spread of the diversity within the ensemble and leads to a better match with the true clustering.

The k-means algorithm with Euclidean distance (or with the cosine measure in the case of text data) is applied as the base clustering algorithm. By default, $b=25$ (unless otherwise specified), and the number of runs per setting is 25. For the `bVote` algorithm, 10 passes over the algorithm are performed and the aggregated partition $\overline{\mathbf{U}}$ with the minimum MSE value is used with the `JS-ALink` algorithm. Given the large size of the Yahoo! and Lansat datasets, the co-association based algorithms are not used with these datasets due to their $O(n^2)$ time and memory complexities.

4.4. Performance evaluation

To evaluate the quality of the consensus partition extracted by the different consensus algorithms, we use an external measure that is widely applied in related literature: the adjusted Rand index (ARI) [32], which measures the agreement between the consensus partition and the true clustering. The ARI measure is computed as follows. Let the true partition be denoted by $\mathbf{U}^*$, and let $\hat{\mathbf{U}}$ denote the extracted consensus partition (after conversion to a hard partition, in the case of voting-based algorithms). Let n_{lq} denote the number of objects that are in both the l-th cluster of $\mathbf{U}^*$, and the q-th cluster of $\hat{\mathbf{U}}$. Let $n_{l.}$ and $n_{.q}$ denote the number of objects in the l-th cluster of $\mathbf{U}^*$ and the q-th cluster of $\hat{\mathbf{U}}$, respectively. The general form of the index is given by (*index*−*expected index*)/(*maximum index*−*expected index*). The expected value of ARI is zero and its maximum is 1. There is a wide range of values that the ARI can take compared to measures taking values between 0 and 1, thus increasing the sensitivity of the index [32]. The ARI takes the value 0 when the index equals its expected value. It is defined as given below

$$\mathrm{ARI} = \frac{\sum_{l,q}\binom{n_{lq}}{2} - \left[\sum_l \binom{n_{l.}}{2}\sum_q \binom{n_{.q}}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_l \binom{n_{l.}}{2}\sum_q \binom{n_{.q}}{2}\right] + \left[\sum_l \binom{n_{l.}}{2}\sum_q \binom{n_{.q}}{2}\right] \Big/ \binom{n}{2}}. \quad (10)$$

Furthermore, the average normalized mutual information (ANMI) defined in [12] is used as an internal measure between the hard consensus partition and members of the ensemble

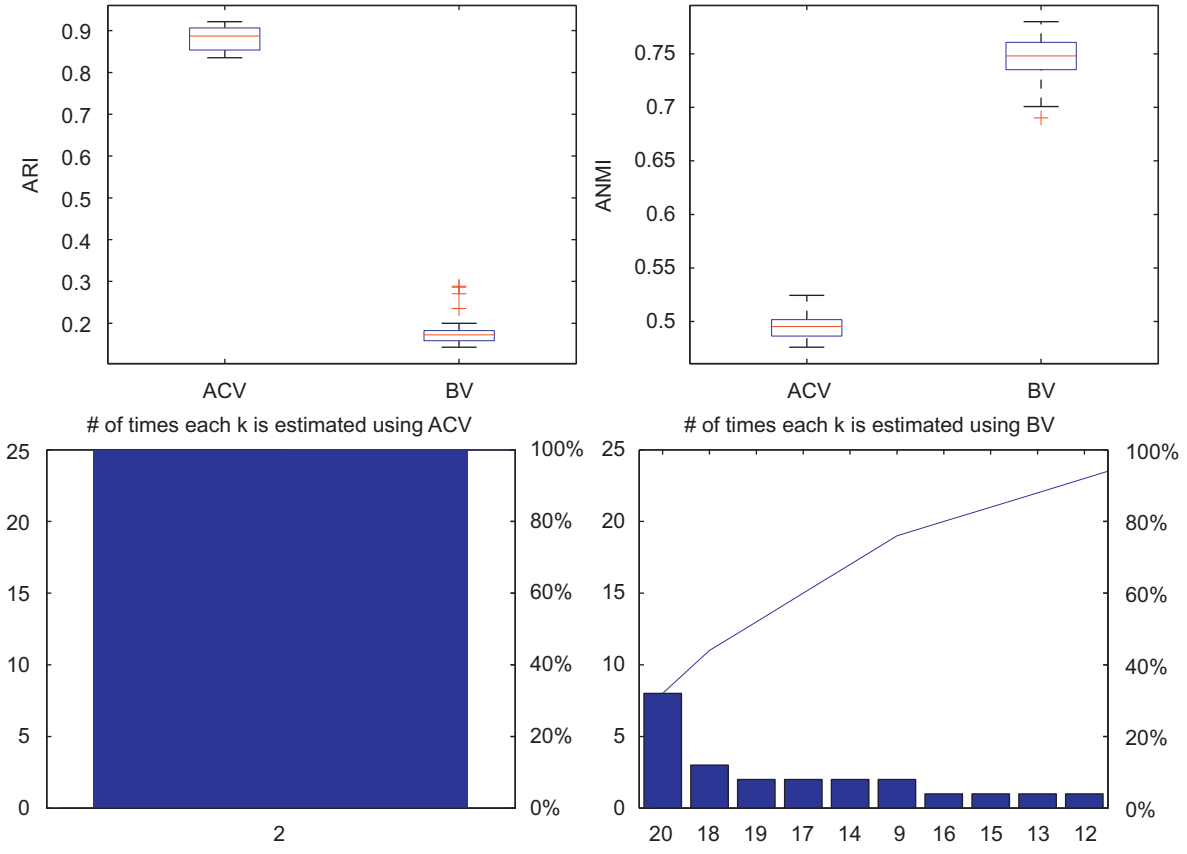

Fig. 1. Results for the 2D2K dataset with $k_i \in [6, 20]$, where k is estimated.

$\{\mathbf{U}^i\}_{i=1}^b$. The normalized mutual information NMI is a measure of the statistical information shared between two clusterings represented as categorical random variables. Let n_{lq}^i denote the number of objects that are in both the l-th cluster of $\hat{\mathbf{U}}^i$, and the q-th cluster of $\hat{\mathbf{U}}$, while $n_{l.}^i$, and $n_{.q}$ denote the number of objects in the l-th cluster of $\mathbf{U}^i$, and the q-th cluster of $\hat{\mathbf{U}}$, respectively. The NMI is defined between $\hat{\mathbf{U}}$, and $\mathbf{U}^i$ below in Eq. (11), and the ANMI is given by $\mathrm{ANMI}(\hat{\mathbf{U}}; \{\mathbf{U}^i\}_{i=1}^b) = (1/b)\sum_{i=1}^b \mathrm{NMI}(\hat{\mathbf{U}}, \mathbf{U}^i)$.

$$\mathrm{NMI}(\hat{\mathbf{U}}, \mathbf{U}^i) = \frac{\sum_{l=1}^{k_i}\sum_{q=1}^{\hat{k}} n_{lq}^i \log\left(\frac{n \times n_{lq}^i}{n_{l.}^i \times n_{.q}}\right)}{\sqrt{\left(\sum_{l=1}^{k_i} n_{l.}^i \log\frac{n_{l.}^i}{n}\right)\left(\sum_{q=1}^{\hat{k}} n_{.q} \log\frac{n_{.q}}{n}\right)}}. \tag{11}$$

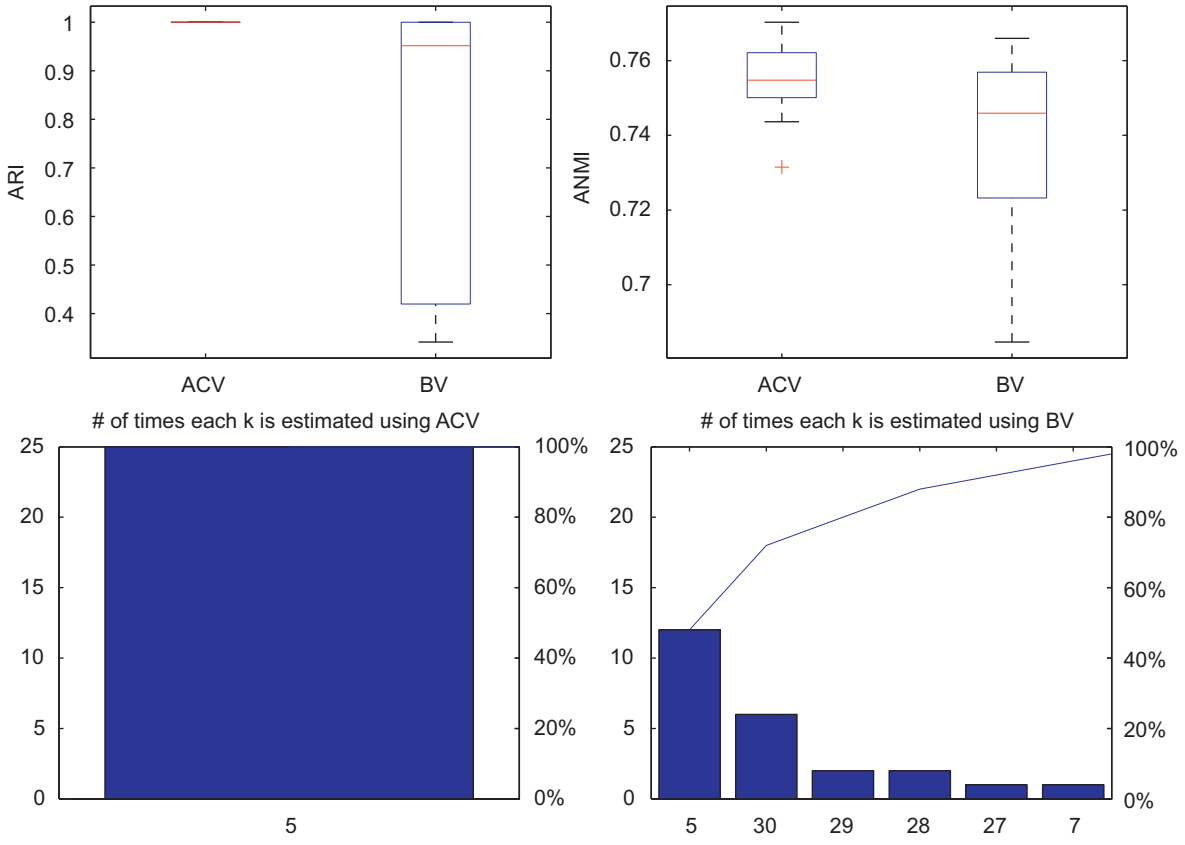

Fig. 2. Results for the 8D5K dataset with $k_i \in [10, 30]$, where k is estimated.

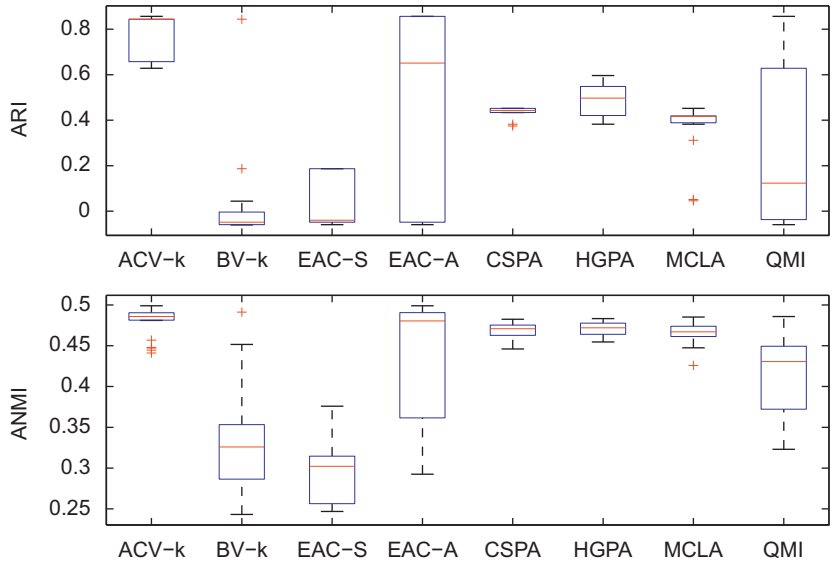

Fig. 3. Results for the two Gauss dataset with predetermined $k = 2$ and $k_i \in [8, 16]$.

4.5. Experimental results

For the first two artificial datasets, where several other consensus algorithms are known to perform well, we focus on comparing ACV versus BV in terms of the quality of their clustering solution including the estimated number of clusters, where the results are presented as follows. The distributions of the ARI and ANMI for ACV versus BV are plotted, and pareto charts depicting the estimated $\hat{k}$ values drawn as bars in descending order of the number of times each value is estimated in 25 runs. The right vertical axis of a pareto chart shows the cumulative percentage of the total number of occurrences of $\hat{k}$. The first 95% of the cumulative distribution is displayed. For the remaining datasets, the results for the consensus partition with a predetermined k are presented, where the distributions of the obtained ARI and ANMI values are shown as box-plots. Furthermore, a comparison of the voting-based consensus algorithms with varying ensemble size is presented.

The results for the 2D2K dataset, where ensembles are generated with $k_i \in [6,20]$, are shown in Fig. 1 where an estimated value for k is computed. As shown in Fig. 1, clustering solutions are highly accurate and estimates of k are perfect using ACV as indicated by the pareto chart ($k=2$ is estimated in 100% of the runs). For the BV algorithm, clustering solutions as well as k are quite poorly estimated. Notably, large values for k are estimated using BV, indicating an inability to extract the global cluster structure inherent in the data. It is further noted that high ANMI values for BV appear to be an effect of its large estimated k values rather than an indication of the quality the clustering solution. In [13], it is observed that the ANMI criterion can be biased toward the average number of clusters in ensemble partitions.

The results for the 8D5K dataset, where ensembles are generated with a variable number of clusters $k_i \in [10,30]$, are presented in Fig. 2 where an estimated value for k is computed. As shown in Fig. 2, clustering solutions and estimates of k are perfect using ACV as indicated by the pareto chart ($k=5$ is estimated in 100% of the runs). On the other hand, poor and highly unstable clustering solutions and less accurate estimates of k are produced using BV. Again, large values for k are estimated using BV, indicating an inability to extract the global cluster structure inherent in the data.

Experimental results comparing all consensus algorithms with predetermined number of clusters are presented in Fig. 3 for the two Gauss dataset, where ensembles are generated with a variable number of clusters $k_i \in [8,16]$. The ACV-k algorithm performs not only substantially better than the BV-k, as indicated by the ARI results, but also it offers significant improvements over all other consensus algorithms, in this case. Again, the ANMI results are not indicative of the clustering quality compared to the true clustering.

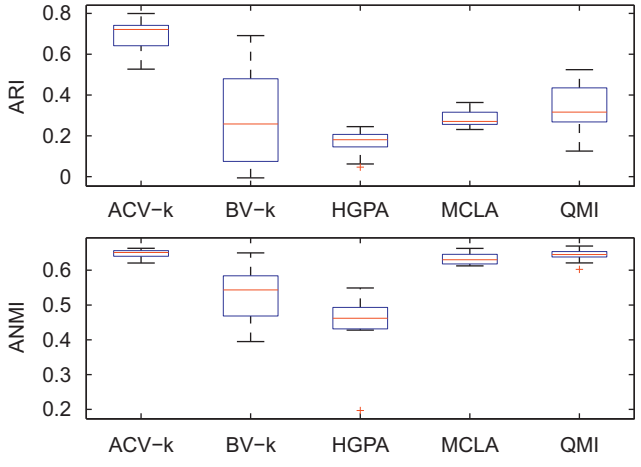

Fig. 4. Results for the Yahoo! dataset with predetermined $k=6$, for $k_i \in [12,24]$.

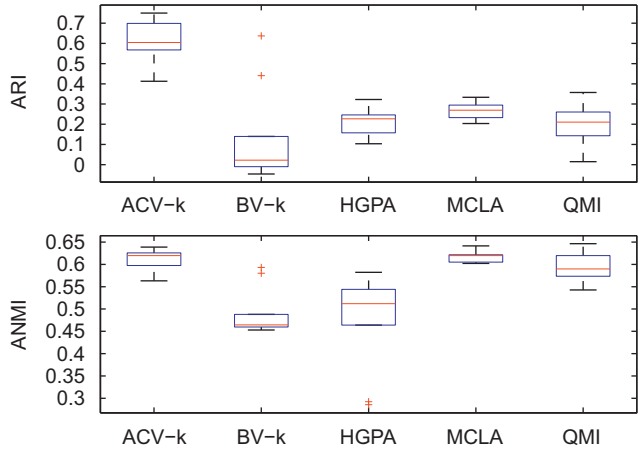

Fig. 5. Results for the Yahoo! dataset with predetermined $k=6$, for $k_i=24$.

For the Yahoo! dataset, a comparison with the most efficient ($O(n)$) consensus algorithms with predetermined number of clusters are presented in Fig. 4 for $k_i \in [12,24]$ and in Fig. 5 for $k_i=24$. As indicated by the ARI values, ACV-k is a winner over all other consensus algorithms, offering significant improvements. Inline with previous experiments, the ANMI results, as an internal measure, do not correlate well with the ARI results.

The results for the Lansat dataset are presented in Fig. 6. From the left plots, it is observed that ACV-k offers substantial improvements over all other consensus algorithms. From the right plots, it is observed that ACV-k and ACV offer substantial improvements over BV-k and BV, for varying b.

The results for the *E. coli* proteins dataset are presented in Fig. 7. From the left plots, it is observed that ACV-k offers substantial improvements over CSPA, HGPA, MCLA, and QMI; and produces competitive clustering quality compared to other better performing algorithms. From the right plots, it is observed that ACV-k and ACV offer improvements, often substantially, over BV-k and BV, for varying ensemble sizes b.

4.6. Conclusion

In this paper, we presented a general formulation for the voting problem, which constitute a key element of the voting-based consensus clustering problem. We presented existing voting schemes, namely the commonly used bipartite matching [15–19], and the cumulative voting scheme [20], as special instances of this general formulation. We considered the aggregated partition generated by a voting-based aggregation algorithm as a statistical representation of the ensemble, on the basis of which an optimally compressed consensus clustering solution can be sought. We further discussed the basis of the `JS-Alink` algorithm described in [20] and applied it in conjunction with each of the bipartite matching and cumulative voting schemes.

In [19], consensus clustering based on bipartite matching was proved to be optimal for ensemble generation techniques in which the ensemble partitions can be viewed as noisy

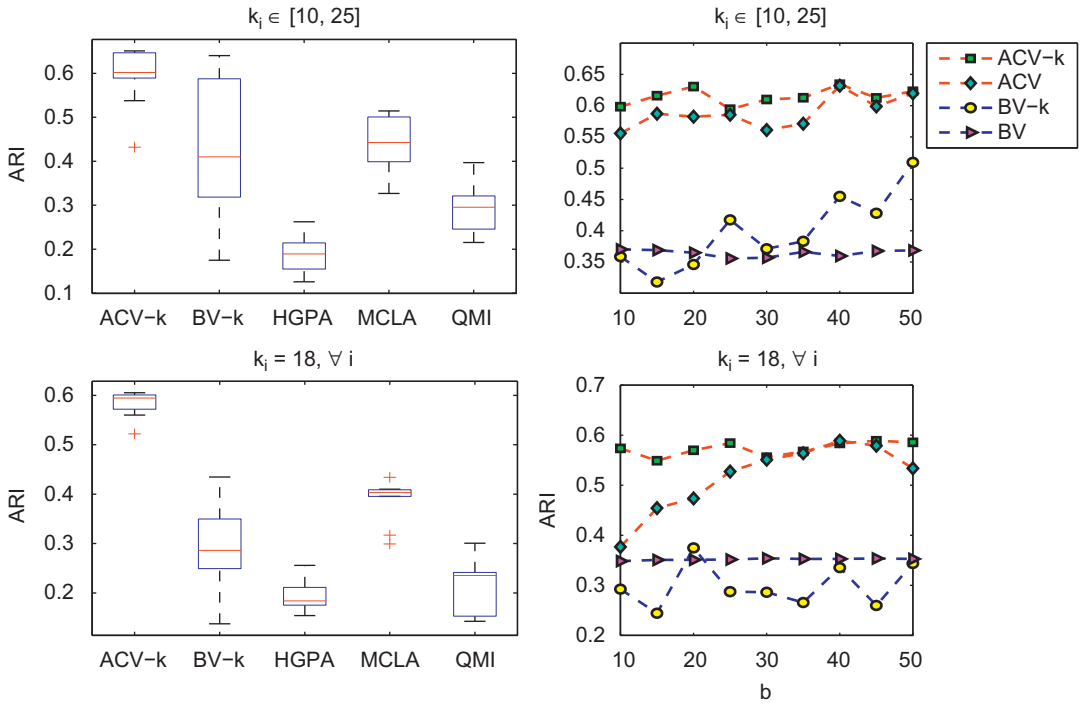

Fig. 6. Results for the Landsat dataset: (left) ACV-k compared to other consensus algorithms with $b=25$ and $k=6$; (right) voting-based consensus algorithms at varying b.

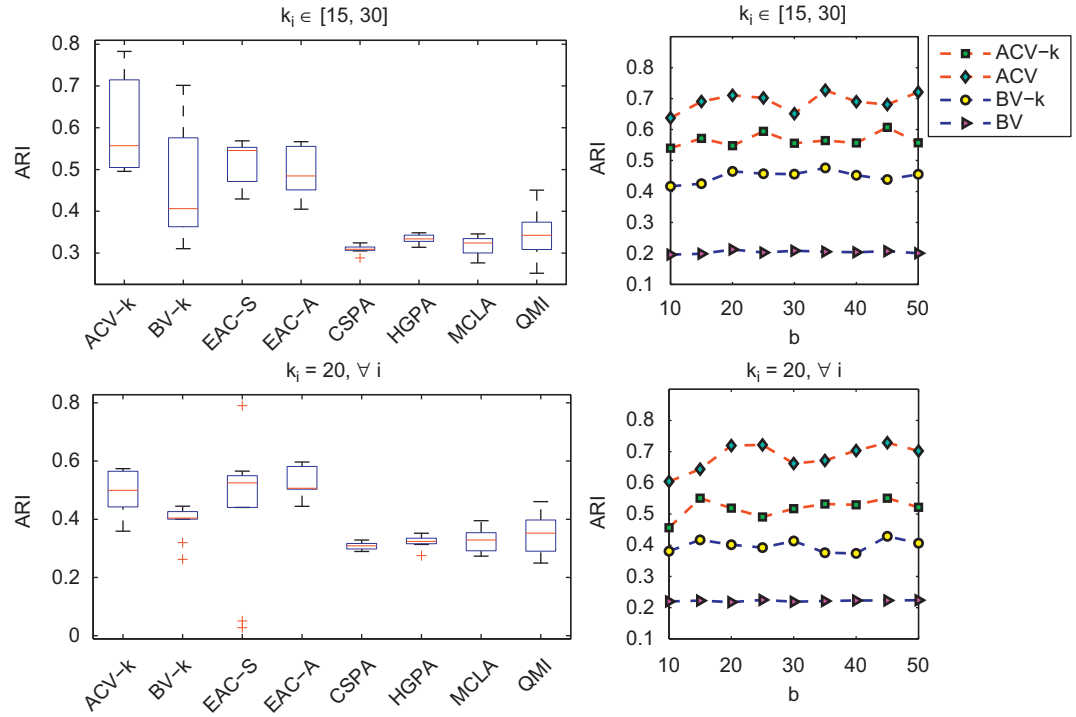

Fig. 7. Results for *E. coli* dataset: (left) ACV-k compared to other consensus algorithms with $b=25$ and $k=8$; (right) voting-based consensus algorithms at varying b.

permutations (with some probability of error) of an underlying clustering that is assumed to be the true clustering. In this paper, we demonstrated that consensus algorithms based on bipartite matching perform very poorly in conjunction with other more general ensemble generation techniques such as those presented in [13,26]. Furthermore, we demonstrated that consensus algorithms based on cumulative voting [20] perform substantially better in this case, compared not only to bipartite matching based algorithms but also to other recent consensus algorithms. Results were presented on three real datasets from different fields of applications. We used the Landsat (Statlog) satellite image dataset; the Yahoo! text data; and the *E. coli* proteins dataset, in addition to three artificial model-based datasets.

We formulated the voting problem as a multi-response regression problem and presented promising empirical results based on the cumulative voting scheme, as a linear regression method. The formulation opens a new direction for researching regression models that can be used more effectively for ensemble-based cluster analysis and for further theoretical investigation.

Another research direction is to investigate the effect of pre-selecting a subset of diverse ensemble partitions on the quality of consensus clustering.

Acknowledgments

The authors would like to thank the anonymous reviewers for their useful comments. This work was supported by the Natural Science and Engineering Research Council (NSERC).

References

[1] J.A. Hartigan, Clustering Algorithms, Wiley, New York, 1975.
[2] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[3] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data, Wiley, New York, 1990.
[4] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (September 1999) 264–323.
[5] C. Fraley, A.E. Raftery, Model-based clustering, discriminant analysis, and density estimation, Technical Report, University of Washington, October 2000.
[6] J.M. Buhmann, Data clustering and learning, in: M. Arbib (Ed.), Handbook of Brain Theory and Neural Networks, MIT Press, Cambridge MA, 2002.
[7] J. Kleinberg, An impossibility theorem for clustering, in: Proceedings of Advances in Neural Information Processing Systems (NIPS), 2002.
[8] N. Tishby, F. Pereira, W. Bialek, The information bottleneck method, in: Proceedings of the 37-th Annual Allerton Conference on Communication Control and Computing, , 1999, pp. 368–377.
[9] Elena D. Cristofor, Information-theoretic methods in clustering, Ph.D. Thesis, University of Massachusetts, 2002.
[10] D.M. Sima, Regularization techniques in model fitting and parameter estimation, Ph.D. Thesis, Faculty of Engineering, K.U. Leuven, Leuven, Belgium, 2006.
[11] J.H. Friedman, Regularized discriminant analysis, Journal of the American Statistical Association 84 (1989)165–175.
[12] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, Journal of Machine Learning Research 3 (2002) 583–617.
[13] A. Fred, A.K. Jain, Combining multiple clusterings using evidence accumulation, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (6) (2005) 835–850.
[14] A. Topchy, A.K. Jain, W. Punch, Clustering ensembles: models of consensus and weak partitions, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (12) (2005) 1866–1881.
[15] E. Dimitriadou, A. Weingessel, K. Hornik, A combination scheme for fuzzy clustering, International Journal of Pattern Recognition and Artificial Intelligence 16 (7) (2002) 901–912.
[16] A.D. Gordon, M. Vichi, Fuzzy partition models for fitting a set of partitions, Psychometrika 66 (2) (2001) 229–248.
[17] S. Dudoit, J. Fridlyand, Bagging to improve the accuracy of a clustering procedure, Bioinformatics 19 (9) (2003) 1090–1099.
[18] B. Fischer, J.M. Buhmann, Bagging for path-based clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (11) (2003) 1411–1415.
[19] A. Topchy, M. Law, A.K. Jain, A. Fred, Analysis of consensus partition in clustering ensemble, in: Proceedings of the IEEE International Conference on Data Mining 2004, Brighton, UK, 2004, pp. 225–232.
[20] H.G. Ayad, M.S. Kamel, Cumulative voting consensus method for partitions with a variable number of clusters, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (1) (2008) 160–173.
[21] Kurt Hornik, A clue for cluster ensembles, Technical Report, Department of Statistics and Mathematics Wirtschaftsuniversität Wien, May 2005.
[22] J.D. Banfield, A.E. Raftery, Model-based Gaussian and non-Gaussian clustering, Biometrics 49 (3) (1993) 803–821.
[23] C. Fraley, A.E. Raftery, How many clusters? which clustering method? answers via model-based cluster analysis, Technical Report 329, Department of Statistics, University of Washington, 1995.
[24] J.C. Dunn, A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters, Journal of Cybernetics 3 (1973) 32–57.
[25] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum, New York, 1981.
[26] L.I. Kuncheva, S.T. Hadjitodorov, Using diversity in cluster ensembles, in: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, The Hague, The Netherlands, 2004, pp. 1214–1219.
[27] H. Kuhn, The Hungarian method for the assignment problem, Naval Research Logistic Quarterly 2 (1955) 83–97.
[28] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data mining, Inference and Prediction, Springer, Berlin, 2001.
[29] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley, New York, USA, 1991.
[30] N. Slonim, N. Tishby, Agglomerative information bottleneck, in: NIPS, 1999, pp. 617–623.
[31] A. Topchy, A.K. Jain, W. Punch, Combining multiple weak clusterings, in: Proceedings of the IEEE International Conference on Data Mining 2003, Melbourne, FL, November 2003, pp. 331–338.
[32] P. Arabie, L. Hubert, Comparing partitions, Journal of Classification 2 (1985) 193–218.
[34] P. Horton, K. Nakai, A probabilistic classification system for predicting the cellular localization sites of proteins, in: Proceeding of the Fourth International Conference on Intelligent Systems for Molecular Biology, 1996, pp. 109–115.
[35] 〈http://archive.ics.uci.edu/ml/〉.

About the Author—HANAN G. AYAD received the Ph.D. degree from the Department of Electrical and Computer Engineering and the M.A.Sc. from the Department of Systems Design Engineering, at the University of Waterloo. She was with the Pattern Analysis and Machine Intelligence research group. She received the B.Sc. degree in Computer Science from the University of Alexandria, Egypt. She received the NSERC Postgraduate Scholarship from the Natural Sciences and Engineering Research Council, the President's Graduate Scholarship from the University of Waterloo, the Ontario Graduate Scholarship, and the Graduate Incentive Award (University of Waterloo). Her research interests are in pattern recognition, machine learning, and information theory, particularly cluster analysis, consensus clustering, ensemble methods, and text mining. She teaches as adjunct lecturer at the University of Waterloo.

About the Author—MOHAMED S. KAMEL received the B.Sc. (Hons) EE (Alexandria University), M.A.Sc. (McMaster University), Ph.D. (University of Toronto). He joined the University of Waterloo, Canada, in 1985 where he is at present Professor and Director of the Pattern Analysis and Machine Intelligence Laboratory at the Department of Electrical and Computer Engineering. Professor Kamel holds Canada Research Chair in Cooperative Intelligent Systems. Dr. Kamel's research interests are in computational intelligence, pattern recognition, machine learning and cooperative intelligent systems. He has authored and co-authored over 350 papers in journals and conference proceedings, seven edited volumes, two patents and numerous technical and industrial project reports. Under his supervision, 67 Ph.D. and M.A.Sc. students have completed their degrees. He is the Editor-in-Chief of the International Journal of Robotics and Automation, Associate Editor of the IEEE SMC, Part A, Pattern Recognition Letters, Cognitive Neurodynamics journal. He is also member of the editorial advisory board of the International Journal of Image and Graphics and the Intelligent Automation and Soft Computing journal. He also served as Associate Editor of Simulation, the Journal of the Society for Computer Simulation. Based on his work at the NCR, he received the NCR Inventor Award. He is also a recipient of the Systems Research Foundation Award for outstanding presentation in 1985 and the ISRAM best paper award in 1992. In 1994 he has been awarded the IEEE Computer Society Press outstanding referee award. He was also a coauthor of the best paper in the 2000 IEEE Canadian Conference on electrical and Computer Engineering. Dr. Kamel is recipient of the University of Waterloo outstanding performance award, the faculty of engineering distinguished performance award. Dr. Kamel is member of ACM, PEO, Fellow of IEEE, and Fellow of EIC. He served as consultant for General Motors, NCR, IBM, Northern Telecom and Spar Aerospace. He is member of the board of directors and co-founder of Virtek Vision Inc. of Waterloo.