BordaConsensus: a New Consensus Function for Soft Cluster Ensembles

Xavier Sevillano, Francesc Alías and Joan Claudi Socoró
GPMM - Grup de Recerca en Processament Multimodal
Enginyeria i Arquitectura La Salle. Universitat Ramon Llull
Quatre Camins, 2 - 08022 Barcelona, Spain
{xavis,falias,jclaudi}@salle.url.edu

ABSTRACT

Consensus clustering is the task of deriving a single labeling by applying a consensus function on a cluster ensemble. This work introduces BordaConsensus, a new consensus function for soft cluster ensembles based on the Borda voting scheme. In contrast to classic, hard consensus functions that operate on labelings, our proposal considers cluster membership information, thus being able to tackle multiclass clustering problems. Initial small scale experiments reveal that, compared to state-of-the-art consensus functions, BordaConsensus constitutes a good performance vs. complexity trade-off.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text Analysis*; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

Document clustering, soft cluster ensembles, Borda voting

1. INTRODUCTION

Being the unsupervised counterpart of classifier committees, consensus clustering aims to combine the results of several clustering processes, collected in the cluster ensemble $\mathbf{\Lambda}$, into a consensus labeling λ through the application of a consensus function $\mathcal{F}$. Typical applications include clustering reuse besides distributed and robust clustering [6].

Most consensus functions posed in the literature operate on *hard* cluster ensembles made up of the *labelings* output by several clustering processes (e.g. [3, 5, 6]). However, they are also applicable on ensembles built upon soft partitioning algorithms, by previously transforming clusterings into labelings (i.e. assigning each document to the cluster with the largest membership probability [4]). In either case, cluster membership information is ignored or discarded.

Alternatively, soft consensus functions have been proposed [2] so as to make use of such information, besides making multiclass clustering consensus possible. Following this approach, this work *i)* adapts existing hard consensus functions to the soft cluster ensembles context, and *ii)* introduces a novel consensus function for soft cluster ensembles, named BordaConsensus, which is inspired in data fusion techniques based on the Borda voting scheme [1]. In order to evaluate our proposal, the BordaConsensus function is compared to state-of-the-art consensus functions in terms of performance and time complexity.

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.
ACM 978-1-59593-597-7/07/0007.

2. THE BORDACONSENSUS FUNCTION

In this section, we describe how the *Borda-fuse* data fusion technique [1] is adapted to derive a novel consensus function for creating a consensus labeling upon soft cluster ensembles.

2.1 Soft cluster ensembles

Henceforth, it is assumed that our aim is to group a collection of $|\mathcal{D}|$ documents into $|\mathcal{C}|$ clusters. Given a set of $|\mathcal{P}|$ independent soft clustering processes, a soft cluster ensemble is defined as a $|\mathcal{C}||\mathcal{P}| \times |\mathcal{D}|$ matrix $\mathbf{\Lambda}$ made up of $|\mathcal{P}|$ membership probability matrices:

$$\mathbf{\Lambda} = \left(\mathbf{M}_1^T \ \mathbf{M}_2^T \ \ldots \ \mathbf{M}_p^T \ \ldots \ \mathbf{M}_{|\mathcal{P}|}^T\right)^T \quad (1)$$

where T denotes matrix transposition and $\mathbf{M}_p$ is the $|\mathcal{C}| \times |\mathcal{D}|$ document-to-cluster membership probabilities matrix resulting from the pth soft partitioning of such corpus:

$$\mathbf{M}_p = \left((\mathbf{m}_1^p)^T \ (\mathbf{m}_2^p)^T \ \ldots \ (\mathbf{m}_c^p)^T \ \ldots \ \left(\mathbf{m}_{|\mathcal{C}|}^p\right)^T\right) \quad (2)$$

where $\mathbf{m}_c^p$ is a column vector that contains the membership probabilities of the $|\mathcal{D}|$ documents with respect to the cth cluster according to the pth soft clustering process.

2.2 The BordaConsensus voting scheme

Data fusion (aka metasearch) techniques are designed to fuse the ranked lists of documents returned by distinct search engines, aiming to improve retrieval results. In the literature, metasearch has often been compared to voting, regarding each search engine as a voter and each document, as a candidate [1]. Following this analogy, we adapt the *Borda-fuse* data fusion technique to the consensus clustering problem, where the $|\mathcal{D}|$ documents, the $|\mathcal{C}|$ predefined clusters and the $|\mathcal{P}|$ clustering processes play the role of candidates, elections and voters, respectively.

Table 1: BordaConsensus algorithm

$score = \text{zeros}(|\mathcal{C}|, |\mathcal{D}|)$;
for $c = 1 \dots |\mathcal{C}|$,
  for $p = 1 \dots |\mathcal{P}|$,
    for $d = 1 \dots |\mathcal{D}|$,
      $d_{win} = \text{argmax}_d\,(\mathbf{m}_c^p)$
      $score\,(c, d_{win}) = score\,(c, d_{win}) + (|\mathcal{D}| - d + 1)$
      $\mathbf{m}_c^p(d_{win}) = 0$
    end
  end
end
$\mathbf{M}_\lambda = \text{softmax}\,(score)$ or $\lambda = \text{argmax}_c\,(score)$

Table 2: Document corpora subsets description

Corpus	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{T}_d\rvert$
miniNewsgroups	6	600	3735	99
OHSUMED	11	1100	4705	120

The application of the Borda positional voting scheme to the consensus clustering problem is as follows: firstly, documents are ranked according to their membership probability with respect to each cluster. Then, for each cluster, the top ranked document receives $|\mathcal{D}|$ points, the second ranked document receives $|\mathcal{D}| - 1$ points, and so on. As a result, the BordaConsensus function can indistinctly yield a soft consensus clustering $\mathbf{M}_\lambda$ (a membership probabilities matrix) or a consensus labeling λ (by assigning the documents to the cluster that maximizes their score –see table 1).

To ensure the consistency of the voting process, it is necessary to solve a cluster correspondence problem across the $|\mathcal{P}|$ partitions prior to voting. In this work, this is accomplished through the correlation based approach that maximizes the overlap between corresponding clusters presented in [5].

3. EXPERIMENTS AND DISCUSSION

The following experiments have been conducted on subsets of the miniNewsgroups and OHSUMED collections, giving rise to two single-class balanced clustering problems. Table 2 summarizes the main aspects of both corpora, including their vocabulary size $|\mathcal{V}|$ and the average number of terms per document, $|\mathcal{T}_d|$.

The application of 4 document representation techniques with 49 distinct dimensionalities and the execution of 10 runs of the k-means (KM) algorithm with random centroids initialization on each corpus has given rise to 200 representationally, partitionally and dimensionally-diverse cluster ensembles of sizes $|\mathcal{P}| = \{4, 10, 49\}$. Note that the soft cluster ensemble $\mathbf{\Lambda}$ is built by transforming the document-to-centroid distances returned by KM into membership probability matrices by applying a softmax normalization.

The proposed BordaConsensus function ($\mathcal{F}^{\text{BC}}$) is compared to several state-of-the-art hard consensus functions: Evidence Accumulation ($\mathcal{F}^{\text{EAC}}$) [3], Cluster-Similarity Partitioning Algorithm ($\mathcal{F}^{\text{CSPA}}$), Hyper-Graph Partitioning Algorithm ($\mathcal{F}^{\text{HGPA}}$) and Meta-Clustering Algorithm ($\mathcal{F}^{\text{MCLA}}$) [6]. This comparison is twofold, as it involves not only the classic *hard* version of such consensus functions, but also a *soft* version (i.e. adapted to operate on soft cluster ensembles). Such adaptation is accomplished by substituting the document or cluster similarity matrices that these consensus functions derive from hard cluster ensembles [3, 6] for the probability matrices $\mathbf{\Lambda}^T \cdot \mathbf{\Lambda}$ or $\mathbf{\Lambda} \cdot \mathbf{\Lambda}^T$, respectively. Moreover, $\mathcal{F}^{\text{BC}}$ is also compared to the only –to our knowledge– consensus function originally designed to operate on soft cluster ensembles, that we name $\mathcal{F}^{\text{DWH}}$ after its authors [2]. This consensus function is based on simultaneously matching clusters iteratively on a pairwise basis plus document-to-cluster assignment according to a proportional weighting voting strategy [2].

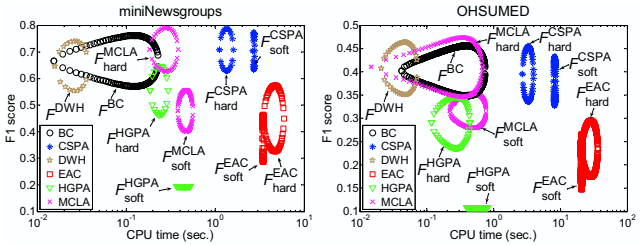

Figure 1: F1 score vs. CPU time 2σ-region plot comparing all the consensus functions.

The experiments have been run under Matlab 7.1 on a PC PIV (1.6 Ghz, 1GB RAM). The final hard consensus labeling λ is evaluated in terms of *i)* the macroaveraged F1 measure with respect to the *true* category labels of each document, and *ii)* the CPU time (in seconds) required for execution.

Figure 1 depicts a F1 score vs. CPU time 2σ-region plot comparing all the consensus functions. Two issues must be noted for both corpora: firstly, the state-of-the-art hard consensus functions suffer F1 score losses when they operate on soft cluster ensembles. And secondly, $\mathcal{F}^{\text{DWH}}$ is, in general, the fastest function, as it simultaneously performs cluster correspondence solving and voting [2]. Regarding the proposed $\mathcal{F}^{\text{BC}}$ function, *i)* in the miniNewsgroups experiment, its performance is statistically significantly better than $\mathcal{F}^{\text{DWH}}$ –in terms of ANOVA ($F\,(1, 398) = 4.36, p = .0374$)– while being clearly faster than the remaining functions, and *ii)* in the OHSUMED experiment, $\mathcal{F}^{\text{BC}}$ achieves the best soft consensus performance (but statistically equivalent to $\mathcal{F}^{\text{DWH}}$), thus constituting a good F1 vs. CPU time trade-off among the soft consensus functions.

Further research will be oriented towards applying the proposed $\mathcal{F}^{\text{BC}}$ function on multiclass consensus clustering problems and implementing other voting schemes.

4. REFERENCES

[1] J.-A. Aslam and M. Montague. Models for metasearch. In *Proc. of the 24th ACM SIGIR Conference*, pages 276–284. 2001.

[2] E. Dimitriadou, A. Weingessel, and K. Hornik. A combination scheme for fuzzy clustering. *International Journal of Pattern Recognition and Artificial Intelligence*, 16(7):901–912, 2002.

[3] A. Fred and A. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005.

[4] A. Jain, M. Murty, and P. Flynn. Data clustering: a survey. *ACM Computing Surveys*, 31(3):264–323, 1999.

[5] S. Siersdorfer and S. Sizov. Restrictive clustering and metaclustering for self-organizing document collections. In *Proc. of the 27th ACM SIGIR Conference*, pages 226–233. 2004.

[6] A. Strehl and J. Ghosh. Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal on Machine Learning Research*, 3:583–617, 2002.