Efficient Ensemble Methods for Document Clustering

Derek Greene, Pádraig Cunningham

University of Dublin, Trinity College,
Dublin 2, Ireland

Abstract. Recent ensemble clustering techniques have been shown to be effective in improving the accuracy and stability of standard clustering algorithms. However, an inherent drawback of these techniques is the computational cost of generating and combining multiple clusterings of the data. In this paper, we present an efficient kernel-based ensemble clustering method suitable for application to large, high-dimensional datasets such as text corpora. To decrease the time required to generate the ensemble members, we employ a *prototype reduction* scheme that makes use of a density-biased selection strategy to construct a smaller kernel matrix that represents a good proxy for the original data. Evaluations performed on text data demonstrate that this process leads to a significant decrease in running time, while maintaining high clustering accuracy.

1 Introduction

Ensemble techniques have been successfully applied in supervised learning to improve the accuracy and stability of classification algorithms, where the rationale is that the combined judgement of a group of predictors is superior to that of an individual (Breiman, 1996). In contrast, cluster analysis methods have often involved the repeated execution of a clustering procedure, followed by the manual selection of an individual solution that maximises a user-defined criterion. However, rather than merely selecting a "winning" partition, recent work has shown that combining the strengths of an ensemble of clusterings can often yield better results (*e.g.* Fred, 2001; Strehl & Ghosh, 2002). Given a collection of clusterings generated on data originating from the same source, the primary aim of *ensemble clustering* is to aggregate the information provided by the collection to produce a more accurate clustering of the data. Additionally, ensemble methods can often afford greater *stability*, which refers to the ability of a clustering procedure to consistently produce similar solutions across multiple trials. Although the underlying "base" clustering algorithm, such as standard k-means with random initialisation, may produce many different partitions of the data of varying accuracy, by combining these partitions we can produce a single definitive solution.

Unfortunately, an inherent drawback of unsupervised ensemble techniques is the computational cost of generating and combining a large number of clusterings of the same data. This is particularly problematic for large, high-dimensional datasets such as text corpora. While reducing the number of ensemble members appears to be a natural solution, an ensemble consisting of too few members is likely to result in an unstable solution that is little better than that produced by the base clustering algorithm.

Greene & Cunningham (2006a) recently proposed an efficient approach for stability-based validation suitable for the task of estimating the number of clusters in large datasets, which is based on the use of a novel prototype reduction scheme. It is apparent that an issue common to both stability analysis and ensemble clustering is the requirement to produce a large, diverse collection of base clusterings. In this paper, we seek to expand upon that work by showing that the principles underlying the reduction technique may also be relevant in improving the efficiency of other computationally costly learning methods. Specifically, we propose a complete ensemble learning process for document clustering that applies correspondence-based aggregation in conjunction with kernel clustering on a matrix constructed using density-biased prototype selection.

The remainder of this paper is organised as follows. The next section provides a summary of relevant work relating to ensemble clustering and prototype reduction. In Section 3 we discuss our proposed clustering scheme, with a particular focus on its application to text data. To demonstrate the effectiveness of the scheme, Section 4 provides comparisons to existing standard and ensemble clustering methods on real-world text datasets. These experiments show that the reduced ensemble clustering process leads to a significant decrease in running time, while maintaining high clustering accuracy.

2 Related Work

2.1 Ensemble Clustering

Ensemble clustering is based on the idea of combining multiple clusterings of a given dataset $\mathcal{X} = \{x_1, \ldots, x_n\}$ to produce a superior aggregated solution. These techniques generally follow a process as illustrated in Figure 1, which consists of two distinct phases:

1. **Generation:** Construct a collection of τ base clustering solutions, denoted as $\mathbb{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_\tau\}$, which represents the members of the ensemble. This is typically done by repeatedly applying a given clustering algorithm in a manner that leads to diversity among the members.
2. **Integration:** Once a collection of ensemble members has been generated, a suitable integration function is applied to combine them to produce a final "consensus" clustering $\bar{\mathcal{C}}$:

   $$f : \{\mathcal{C}_1, \ldots, \mathcal{C}_\tau\} \rightarrow \bar{\mathcal{C}}$$

   In practice, this often involves the application of an additional clustering procedure to an intermediate representation of $\mathbb{C}$.

We now summarise the most popular generation and integration techniques that have been proposed in recent literature.

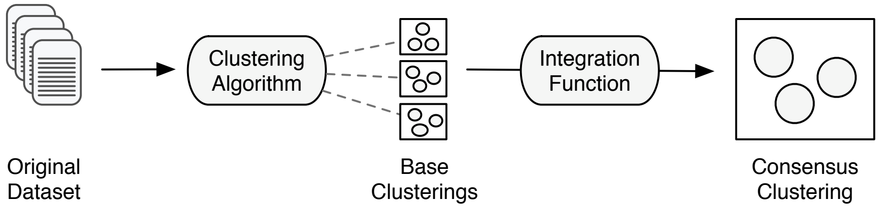

Fig. 1. Generic ensemble clustering process.

Generation. It has frequently been demonstrated that supervised ensembles are most successful when constructed from a set of accurate classifiers whose errors lie in different parts of the data space (*e.g.* Opitz & Shavlik, 1996). Similarly, unsupervised ensemble procedures typically seek to encourage diversity with a view to improving the quality of the information available in the integration phase. A variety of strategies have been proposed to achieve this goal, including random initialisation (Jain & Fred, 2002), feature extraction based on random projection (Fern & Brodley, 2003), random parameter selection (Jain & Fred, 2002) and the combination of sets of "weak" partitions (Topchy *et al.*, 2003). However, by far the most commonly used strategy has been to employ unbiased random sampling to produce partitions on different subsets of the same dataset. Many authors (*e.g.* Leisch, 1999; Dudoit & Fridlyand, 2003) have suggested the use of a bootstrapping aggregation or "bagging" technique, where subsets of the original data are produced by independently drawing with replacement. A related technique involves applying subsampling without replacement, where typically 60-80% of the data objects are used when generating each base clustering (Minaei-Bidgoli *et al.*, 2004; Fern & Brodley, 2004). Having chosen a sample of data, an ensemble member is generated by applying a suitable base clustering algorithm, such as standard k-means with random initialisation.

Integration. In supervised learning, it has been observed that the success of an ensemble technique depends not only on the presence of a diverse set of base classifiers, but also on the ability of the integration method to exploit the resulting diversity (Brodley & Lane, 1996). Similarly, the choice of a suitable method for combining an ensemble of clusterings will greatly affect the accuracy of the final clustering solution (Greene *et al.*, 2004).

Several different approaches have been proposed for performing the task of integration. The most popular has been to use the information provided by the ensemble members to derive a new measure of similarity between data objects. This information may be represented in the form of a pairwise co-association matrix, which can be subsequently used by a similarity-based clustering algorithm to produce a final partition of the data (Fred, 2001). Rather than examining pairwise associations between objects, several authors have suggested analysing the relationships between the individual clusters contained in all partitions in $\mathbb{C}$. These relationships may be modelled in the form of a weighted hypergraph

(Strehl & Ghosh, 2002) or a bipartite graph (Fern & Brodley, 2004), which may subsequently be divided using popular graph partitioning or spectral clustering techniques.

The graph-based approach proposed by Strehl & Ghosh (2002) is based on the assumption that there will be a direct relationship between individual clusters across different partitions in $\mathbb{C}$. This concept of correspondence has been explicitly used by several authors for combining collections of clusterings. Dudoit & Fridlyand (2003) proposed a method, referred to as *BagClust1*, which involves aligning the clusters in each newly generated base partition with those in the existing ensemble clustering. The new cluster assignments may then be viewed as votes indicating the strength of assignment of an object to each of the k clusters in the current ensemble. Once all ensemble members have been added, a final clustering is obtained by taking the majority cluster for each object. Note that this approach assumes that each partition contains a fixed number of clusters k.

2.2 Prototype Reduction

Prototype reduction has been extensively used in supervised learning to improve the efficiency of learning tasks involving large datasets. These techniques are concerned with producing a minimal set of objects or prototypes to represent the data, while ensuring that a classifier applied to this set will perform approximately as well as on the original dataset. In the literature, reduction techniques are generally divided into two broad categories: *prototype selection* techniques seek to identify a subset of representative objects from the original data, while *prototype extraction* techniques involve the creation of an entirely new set of objects. A variety of supervised reduction schemes have been proposed in the literature (see Bezdek & Kuncheva, 2001). One particularly novel technique, described by Kim & Oommen (2005), involves using a standard reduction scheme to produce a reduced set of prototypes, from which a smaller kernel matrix is constructed. Ensemble classifier methods are then employed using this matrix to compensate for any loss in accuracy resulting from the application of the reduction procedure.

While most work in prototype reduction has focused on supervised learning tasks, the concept has also been used to improve the efficiency of cluster analysis procedures. Greene & Cunningham (2006a) proposed an unsupervised kernel-based reduction scheme, where new prototypes are formed by locally combining subsets of the original dataset. Specifically, n extracted prototypes may be potentially constructed by finding the mean of each object together with its set of p nearest neighbours. From these, a subset of $n' < n$ prototypes are selected using a *density-biased* selection strategy to ensure that all cluster structures in the data are adequately represented. Rather than computing explicit representations for the new prototypes in the original feature space, the values in the full $n \times n$ kernel matrix $\mathbf{K}$ are used to directly construct a reduced $n' \times n'$ matrix $\mathbf{K}'$. This representation is subsequently used to decrease the computational expense of performing stability-based validation.

3 Proposed Method

As noted previously, a significant disadvantage of ensemble techniques is the computational cost of repeatedly generating and combining partitions of a given dataset. In particular, the feasibility of applying popular techniques such as those described in Section 2.1 may be greatly limited by the number of objects n. The number of feature m used to represent the objects can also limit their usefulness when working with high-dimensional data, such as document collections.

3.1 Kernel-Based Ensemble Clustering

To avoid having to repeatedly recompute similarity values in the original feature space, we choose to represent the data in the form of an $n \times n$ kernel matrix $\mathbf{K}$, where K_{ij} indicates the affinity between objects x_i and x_j. The advantage of using kernel methods in the context of ensemble clustering derives from the fact that, having constructed a single kernel matrix, we may subsequently generate multiple partitions without referring back to the original data. The standard k-means algorithm with cosine similarity has commonly been used in document clustering. Therefore, an intuitive choice for a base clustering algorithm is to make use of the corresponding kernelised k-means algorithm applied to a normalised linear kernel (Schölkopf & Smola, 2001). However, while this kernel represents a suitable choice for document clustering, like many kernel functions applied to sparse data, its matrix will often suffer from the problem of *diagonal dominance*. This phenomenon occurs when, for a given kernel function, self-similarity values are large relative to between-object similarities. This can negatively impact upon the accuracy and stability of centroid-based kernel clustering algorithms. To address this issue, we make use of kernel k-means with *algorithm adjustment* as described by Greene & Cunningham (2006b).

To encourage diversity among the ensemble members, we apply subsampling without replacement and apply randomly-initialised kernel clustering to the selected rows of the kernel matrix. Minaei-Bidgoli *et al.* (2004) demonstrated that ensembles created in this way can lead to results that are comparable to bootstrap aggregation, while requiring less computational time to produce the base clusterings. We have observed similar behaviour when this generation approach is applied to text data, where we employ a sampling factor $\beta = 0.8$. After each subsampling is partitioned, we produce a clustering of all n objects by applying a classification scheme to predict memberships for the out-of-sample objects. This is similar to the approach used in prediction-based validation (Tibshirani *et al.*, 2001), where a classifier is selected so as to “mimic” the behaviour of the clustering algorithm. In this context, we apply a kernel nearest centroid prediction method, where each missing object is assigned to the most similar pseudo-centroid in the base clustering.

Once a collection of base clusterings $\mathbb{C}$ has been generated, we integrate the collection by employing a correspondence clustering technique similar to the *Bag-Clust1* algorithm proposed by Dudoit & Fridlyand (2003). Unlike other ensemble clustering schemes, the final clustering of the data is constructed incrementally

```
1. Construct full kernel matrix $\mathbf{K}$ and set counter $t = 0$.
2. Increment $t$ and generate base clustering $\mathcal{C}_t$:
     (i) Produce a subsampling without replacement.
    (ii) Apply adjusted kernel $k$-means with random initialisation to the samples.
   (iii) Assign each out-of-sample object to the nearest centroid in $\mathcal{C}_t$.
3. If $t = 1$, initialise $\mathbf{V}$ as the $n \times k$ binary membership matrix for $\mathcal{C}_1$.
   Otherwise, update $\mathbf{V}$ as follows:
     (i) Compute the current consensus clustering $\bar{\mathcal{C}}$ from $\mathbf{V}$ such that
             $x_i \in \bar{C}_j \;\text{ if }\; j = \arg\max_j V_{ij}$
    (ii) Find the optimal correspondence $\pi(\mathcal{C}_t)$ between the clusters in $\mathcal{C}_t$ and $\bar{\mathcal{C}}$.
   (iii) For each object $x_i$ assigned to the $j$-th cluster in $\pi(\mathcal{C}_t)$, increment $V_{ij}$.
4. Repeat from Step 2 until $\bar{\mathcal{C}}$ is stable or $t = \tau_{max}$.
5. Return the final consensus clustering $\bar{\mathcal{C}}$.
```

Fig. 2. Kernel-based correspondence clustering.

as each ensemble member is generated, so that we do not require the application of a subsequent clustering procedure to produce a final solution. Additionally, this scheme avoids the large storage overhead of maintaining an intermediate representation of the collection $\mathbb{C}$, which is a notable drawback of graph-based integration schemes. In practice, we observe that correspondence-based integration produces more stable results than other schemes such as those based on pairwise co-assignment, which are often highly sensitive to the choice of final clustering algorithm (Greene *et al.*, 2004).

The kernel-based correspondence clustering scheme proceeds as summarised in Figure 2. Having generated the first ensemble member $\mathcal{C}_1$, a $n \times k$ membership matrix $\mathbf{V}$ is constructed such that:

$$V_{ij} = \begin{cases} 1 & \text{if } x_i \in C_j \text{ in } \mathcal{C}_1 \\ 0 & \text{otherwise.} \end{cases}$$

As each subsequent clustering $\mathcal{C}_t$ is generated, the values in $\mathbf{V}$ are updated. Unlike when combining classifiers, the clusters in each partition will not have a pre-defined label. Therefore, each new set of clusters must be aligned with those that have been previously generated. The current consensus clustering $\bar{\mathcal{C}}$ is computed by taking the majority cluster label for each object based on the row values in $\mathbf{V}$. We then find the best match between those clusters and the existing clusters in $\mathcal{C}_t$. The optimal permutation $\pi(\mathcal{C}_t)$ may be found in $O(k^3)$ time by solving the minimal weight bipartite matching problem using the Hungarian method (Kuhn, 1955). For each object x_i assigned to the j-th cluster in $\pi(\mathcal{C}_t)$, we then increment the entry V_{ij}. When all ensemble members have been generated, $\bar{\mathcal{C}}$ represents the final consensus clustering of the data.

An issue that is often overlooked in ensemble clustering is the choice of a suitable value for the number of ensemble members τ. If τ is too large, the running time of the ensemble process will be prohibitive. On the other hand, if τ

is too small, it is likely that the final ensemble solution will be unstable due to the stochastic nature of the generation scheme. One benefit of the correspondence clustering approach is that, by performing the integration process in parallel with the generation phase, we may easily determine whether the ensemble process may be terminated. Specifically, we choose to automatically stop generating ensemble members when the the cluster assignments in $\bar{\mathcal{C}}$ remain unchanged for a fixed number of generations. The process may also be terminated if the number of members t reaches a pre-defined maximum value τ_{max}.

3.2 Ensemble Clustering with Kernel Reduction

The ensemble clustering approach introduced in Section 3.1 allows each base clustering to be generated without referring back to the original feature space. However, for larger datasets, the computational cost of repeatedly applying an algorithm requiring $O((\beta n)^2)$ time may still be prohibitive. Clearly, decreasing n would make the ensemble process significantly less computationally expensive. Therefore, we now expand upon the work described by Greene & Cunningham (2006a), showing that the principles underlying the kernel-based prototype reduction technique may also be used to greatly improve the efficiency of ensemble clustering. Briefly, the proposed techniques involves applying prototype reduction, performing correspondence clustering on the reduced representation and subsequently mapping the resulting aggregate solution back to the original data. An outline of the entire process is illustrated in Figure 3.

The initial reduction process follows that described in Greene & Cunningham (2006a). Firstly, the original $n \times n$ kernel matrix $\mathbf{K}$ is transformed to a condensed $n' \times n'$ matrix $\mathbf{K}'$, where $n' = \frac{n}{\rho}$ and ρ is a user-defined parameter controlling the reduction rate. Specifically, n extracted prototypes may be potentially constructed by finding the mean of each object together with its set of p nearest neighbours. From these, a subset of $n' < n$ prototypes are selected using a *density-biased* selection strategy. The matrix $\mathbf{K}'$ may be directly constructed from the affinity values in $\mathbf{K}$ without referring back to the original feature space. In practice, we use a reduction rate of $\rho = 4$ and consider prototypes constructed from small, homogenous neighbourhoods ($p = 5$), as these parameter values were previously shown to be useful for a range of text datasets.

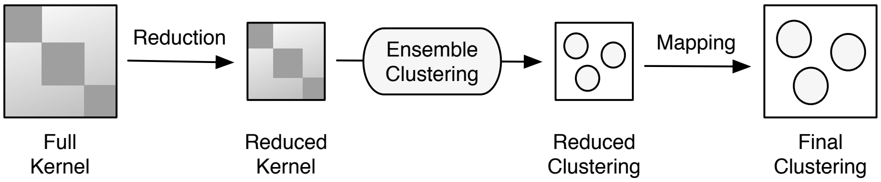

Fig. 3. Ensemble clustering process with prototype reduction.

1. Construct full $n \times n$ kernel matrix $\mathbf{K}$ from the original data $\mathcal{X}$.
2. Apply prototype reduction to form the $n' \times n'$ reduced kernel matrix $\mathbf{K}'$.
3. Apply kernel-based correspondence clustering using $\mathbf{K}'$ as given in Figure 2 to produce a consensus clustering $\bar{\mathcal{C}}'$.
4. Construct a full clustering $\hat{\mathcal{C}}$ by assigning a cluster label to each x_i based on the nearest cluster in $\bar{\mathcal{C}}'$.
5. Apply adjusted kernel k-means using $\hat{\mathcal{C}}$ as an initial partition to produce a refined final clustering of $\mathcal{X}$.

Fig. 4. Kernel-based correspondence clustering with prototype reduction.

Once we have constructed the reduced kernel matrix, the ensemble clustering process is performed as given in Figure 2. The application of the proposed reduction strategy results in a significant decrease in the computational cost of this process. When generating each ensemble member, the cost of clustering is reduced to $O((\frac{\beta n}{\rho})^2)$. In addition, the time required to construct a cost matrix for the Hungarian matching method and the time needed to update $\mathbf{V}$ are both decreased to $O(n')$.

After the ensemble process has terminated, the problem remains of deriving a final clustering $\hat{\mathcal{C}}$ of the original n data objects from the consensus clustering of reduced prototypes $\bar{\mathcal{C}}'$. An intuitive way of achieving this is to assign each original object x_i to the nearest centroid in $\bar{\mathcal{C}}'$. Just as each reduced prototype can be decomposed into a set of $p+1$ original objects, we can also decompose the centroid of each reduced cluster into the mean of all the original objects which form the reduced prototypes assigned to that cluster. In practice, we can identify the nearest cluster based on values in the original kernel matrix $\mathbf{K}$ and the list of nearest neighbours used to form the reduced prototypes. This mapping of $\bar{\mathcal{C}}'$ to a clustering of $\mathcal{X}$ can be performed in time $O(n'n)$. To further improve the accuracy of this solution, we suggest a refinement procedure that involves applying adjusted kernel k-means to $\hat{\mathcal{C}}$ using the full matrix $\mathbf{K}$. In practice, we observe that this generally requires very few reassignment iterations, while leading to a noticeable increase in clustering accuracy. The entire ensemble process with prototype reduction is summarised in Figure 4.

4 Evaluation

4.1 Experimental Setup

In order to assess the techniques proposed in Section 3, we conducted a comparison on ten datasets that have previously been used in the evaluation of document clustering algorithms (see Table 1). For further information regarding these collections, consult Greene & Cunningham (2006a). To pre-process the datasets we applied standard stop-word removal and stemming techniques. We subsequently removed terms occurring in less than three documents and applied log-based TF-IDF normalisation to the feature vectors.

Dataset	Description	Documents	Terms	k
bbc	News articles from BBC	2225	9635	5
bbcsport	Sports news articles from BBC	737	4613	5
classic	CISI/CRAN/MED sets	7097	8276	4
classic3	CACM/CISI/CRAN/MED sets	3893	6733	3
cstr	Computer science technical abstracts	505	2117	4
ng17-19	Overlapping newsgroups	2625	12020	3
ng3	Well-separated newsgroups	2928	12357	3
reuters5	Top 5 categories from *Reuters-21578*	2317	4627	5
reviews	Entertainment articles from TREC	4069	18152	5
sports	Sports news articles from TREC	8580	14615	7

Table 1. Details of experimental datasets.

The primary focus of our evaluation was to consider the effects of applying prototype reduction prior to ensemble clustering, in terms of accuracy, stability and running time. Specifically, we compare three variations of correspondence-based ensemble clustering: using standard k-means on the original feature space (COR-KM), adjusted kernel k-means on the full kernel matrix (COR-AA) and adjusted kernel k-means on the reduced kernel matrix (COR-RED). For these techniques, we average the results over 25 trials. In each trial, we automatically terminate the ensemble process after 30 stable iterations have elapsed or when $\tau_{max} = 250$ ensemble members have been generated. As a baseline comparison, we also include two base clustering algorithms: k-means with cosine similarity (KM) and adjusted kernel k-means using a normalised linear kernel (AA). For these experiments, we performed random initialisation and averaged the results over 250 trials to compensate for the inherent instability of both algorithms. In all cases, we set the number of clusters k to correspond to the number of natural classes in the data.

4.2 Comparison of Algorithm Accuracy

To evaluate algorithm accuracy, we employ external validation based on the *normalised mutual information* (NMI) measure (Strehl & Ghosh, 2002). Table 2 summarises the mean and standard deviation of the NMI scores for the five clustering methods under consideration. On all datasets, the kernel-based ensemble techniques lead to an improvement over both base clustering algorithms. These techniques also performed at least as well as correspondence-based ensemble clustering using standard k-means on the original feature space (COR-KM), and frequently achieved higher accuracy. We suggest that the applicable of a diagonal dominance reduction technique, which limits the influence of self-similarity, contributes to this improvement. In addition, the results in Table 2 show that in several cases correspondence clustering after prototype reduction (COR-RED) performed better than clustering on the full kernel matrix. We suggest that the use of neighbourhood centroids as prototypes allows the production of a robust partition that may not be easily obtained by clustering on the full dataset using standard initialisation strategies. The subsequent application of a full cluster-

Dataset	KM	AA	COR-KM	COR-AA	COR-RED
bbc	0.81 ± 0.08	0.85 ± 0.06	**0.88 ± 0.00**	**0.88 ± 0.00**	**0.88 ± 0.00**
bbcsport	0.73 ± 0.10	0.80 ± 0.08	0.87 ± 0.01	**0.90 ± 0.00**	0.89 ± 0.03
classic	0.70 ± 0.04	0.74 ± 0.02	0.69 ± 0.00	**0.75 ± 0.00**	**0.75 ± 0.00**
classic3	0.93 ± 0.08	0.94 ± 0.06	**0.95 ± 0.00**	**0.95 ± 0.00**	**0.95 ± 0.00**
cstr	0.69 ± 0.05	0.74 ± 0.04	0.76 ± 0.01	0.76 ± 0.01	**0.77 ± 0.03**
ng17	0.41 ± 0.12	0.42 ± 0.13	0.47 ± 0.04	0.51 ± 0.05	**0.55 ± 0.04**
ng3	0.83 ± 0.10	0.84 ± 0.10	0.89 ± 0.00	0.90 ± 0.00	**0.91 ± 0.00**
reuters5	0.55 ± 0.07	0.59 ± 0.04	0.60 ± 0.00	**0.61 ± 0.00**	**0.61 ± 0.01**
reviews	0.56 ± 0.08	0.58 ± 0.05	**0.61 ± 0.00**	**0.61 ± 0.00**	**0.61 ± 0.00**
sports	0.62 ± 0.05	0.67 ± 0.06	0.66 ± 0.01	**0.70 ± 0.02**	0.69 ± 0.02

Table 2. Accuracy (NMI) scores for base and ensemble clustering methods.

ing phase allows this partition to be refined to produce a more accurate final solution.

Both KM and AA exhibited considerable instability due to the sensitivity of these algorithms to the choice of initial clusters, which is reflected in the high deviation scores in Table 2. In contrast, the ensemble methods tend to be far more robust, frequently producing identical or highly similar partitions. Only in the case of the *bbcsport* and *cstr* datasets did the ensemble methods suffer any noticeable degradation in stability due to prototype reduction. This is likely to be due to the small size of the datasets, and we suggest that a higher number of ensemble members may be appropriate for smaller text datasets. As the time required to generate each member for small datasets is extremely low, this should not pose a significant problem in practice.

4.3 Comparison of Algorithm Efficiency

Another important aspect of our evaluation was to assess the computational gains resulting from prototype reduction. Table 3 provides a list of the mean running times for the ensemble clustering experiments, which were performed on a Pentium IV 3.4GHz, 2GB RAM running Sun Java 1.5. The cost of the mapping and refinement procedures in COR-RED has the effect that the computational savings are not as dramatic as those observed by Greene & Cunningham (2006a) in stability analysis. However, the gains afforded by working on a reduced kernel matrix are still very significant. Only in the case of the *classic* dataset did reduction fail to significantly reduce computational cost relative to the other ensemble techniques. Note that the application of the early termination technique for correspondence clustering also has a significant influence on the running times in Table 3.

We did observe that the procedures running on the full $n \times n$ kernel matrices (COR-AA) took significantly longer than those performed using standard k-means, particularly as n increases. This results from the fact that the implementation of k-means in our toolkit is optimised to take advantage of the sparse nature of text data. However, we note that this improvement only occurs for datasets whose term-document matrix consists of at least 98% zero values, and

Dataset	COR-KM	COR-AA	COR-RED
bbc	95	215	**13**
bbcsport	27	34	**1**
classic	**101**	6181	157
classic3	32	359	**26**
cstr	7	14	**1**
ng17	85	753	**40**
ng3	57	485	**19**
reuters5	40	395	**26**
reviews	281	1722	**81**
sports	968	17954	**579**

Table 3. Mean running times (in seconds) for ensemble clustering procedures.

is specific to the use of sparse matrix storage and the cosine similarity measure. In contrast, we suggest that the reduced ensemble procedure introduced in this paper may be used to increase the efficiency of ensemble clustering when applied to a wide range of data and when using an arbitrary similarity metric. In addition, further optimisations may be possible for certain types of data when using a sparse kernel matrix representation.

We note that it is possible to further reduce the computational time of ensemble generation by using a smaller factor for subsampling (*e.g.* $\beta = 0.4$). However, as discussed by Minaei-Bidgoli *et al.* (2004), a critical sampling size for a given dataset is required to match the accuracy afforded by more expensive bagging generation strategies. For several datasets, using smaller subsamplings lead to a less accurate consensus clustering and higher instability. Consequently, we suggest that using prototype reduction with a relatively high sampling rate (*i.e.* $\beta = 0.8$) represents a pragmatic choice for providing sufficient diversity and ensuring stability on a range of datasets.

5 Conclusion

In this paper we built upon our previous work by investigating the use of prototype reduction in ensemble clustering. Specifically, we introduced an efficient method for ensemble clustering based on the use of kernel learning methods and density-biased prototype reduction. We evaluated this method on real-world text datasets, where the reduced ensemble clustering process was shown to frequently afford a significant decrease in running time, while maintaining high clustering accuracy. In several cases, the proposed method out-performed more computationally costly ensemble techniques operating on the original data.

While we have applied kernel-based prototype reduction in conjunction with a correspondence clustering scheme, we suggest that the kernel-based prototype reduction may also be useful when employing other ensemble integration schemes, such as those based on analysing pairwise co-assignments. In future, we intend to apply these techniques to other domains where ensemble clustering has previously been applied.

Bibliography

Bezdek, J.C. & Kuncheva, L. (2001). Nearest prototype classifier designs: An experimental study. *International Journal of Intelligent Systems*, **16**, 1445–1473.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, **24**, 123–140.

Brodley, C. & Lane, T. (1996). Creating and exploiting coverage and diversity. In *Proc. AAAI Workshop on Integrating Multiple Learned Models*, 8–14, Portland, Oregon.

Dudoit, S. & Fridlyand, J. (2003). Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, **19**, 1090–1099.

Fern, X. & Brodley, C. (2004). Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of ICML'04*.

Fern, X.Z. & Brodley, C.E. (2003). Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of 20th International Conference on Machine learning (ICML2003)*, Washington.

Fred, A. (2001). Finding consistent clusters in data partitions. In *Proceedings of Multiple Classifier Systems : Second International Workshop (MCS 2001)*, vol. 2096, 309.

Greene, D. & Cunningham, P. (2006a). Efficient prediction-based validation for document clustering. Tech. Rep. CS-2006-22, Trinity College Dublin.

Greene, D. & Cunningham, P. (2006b). Practical solutions to the problem of diagonal dominance in kernel document clustering,. In *Proc. 23rd International Conference on Machine Learning*.

Greene, D., Tsymbal, A., Bolshakova, N. & Cunningham, P. (2004). Ensemble clustering in medical diagnostics. In *CBMS '04: Proceedings of the 17th IEEE Symposium on Computer-Based Medical Systems (CBMS'04)*, 576, IEEE Computer Society, Washington, DC, USA.

Jain, A.K. & Fred, A. (2002). Evidence accumulation clustering based on the k-means algorithm. *Structural, Syntactic, and Statistical Pattern Recognition*, **LNCS 2396**, 442–451.

Kim, S.W. & Oommen, B.J. (2005). On using prototype reduction schemes and classifier fusion strategies to optimize kernel-based nonlinear subspace methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 455–460.

Kuhn, H.W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quaterly*, **2**, 83–97.

Leisch, F. (1999). Bagged clustering. Working Paper 51, SFB "Adaptive Information Systems and Modeling in Economics and Management Science".

Minaei-Bidgoli, B., Topchy, A.P. & Punch, W.F. (2004). A comparison of resampling methods for clustering ensembles. In *IC-AI*, 939–945.

Oehlschlägel, J. (2006). Truecluster: scalable statistical clustering with model selection.

Opitz, D.W. & Shavlik, J.W. (1996). Generating accurate and diverse members of a neural-network ensemble. In *Advances in Neural Information Processing Systems*, vol. 8, 535–541.

Schölkopf, B. & Smola, A.J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA.

Strehl, A. & Ghosh, J. (2002). Cluster ensembles - a knowledge reuse framework for combining partitionings. In *Proc. Conference on Artificial Intelligence (AAAI 2002), Edmonton*, 93–98, AAAI/MIT Press.

Tibshirani, R., Walther, G., Botstein, D. & Brown, P. (2001). Cluster validation by prediction strength. Tech. rep., Statistics Department, Stanford University.

Topchy, A., Jain, A. & Punch, W. (2003). Combining multiple weak clusterings. In *Proc. Third IEEE International Conference on Data Mining (ICDM'03)*, 331–338, Melbourne, Florida.