A Matrix Factorization Approach for Integrating Multiple Data Views

Derek Greene, Pádraig Cunningham

School of Computer Science & Informatics, University College Dublin
{derek.greene,padraig.cunningham}@ucd.ie

Abstract. In many domains there will exist different representations or "views" describing the same set of objects. Taken alone, these views will often be deficient or incomplete. Therefore a key problem for exploratory data analysis is the integration of multiple views to discover the underlying structures in a domain. This problem is made more difficult when disagreement exists between views. We introduce a new unsupervised algorithm for combining information from related views, using a *late integration* strategy. Combination is performed by applying an approach based on matrix factorization to group related clusters produced on individual views. This yields a projection of the original clusters in the form of a new set of "meta-clusters" covering the entire domain. We also provide a novel model selection strategy for identifying the correct number of meta-clusters. Evaluations performed on a number of multi-view text clustering problems demonstrate the effectiveness of the algorithm.

1 Introduction

In many data analysis tasks there will naturally exist several different ways to describe the same set of data objects. This leads to the availability of multiple distinct representations or "views" that encode patterns relevant to the domain [1]. The question then arises, how can we integrate these representations in a way that allows us to effectively identify and explore these patterns? For some data exploration applications, we may have access to a set of views that are entirely *compatible* – the same patterns will occur across all views. The problem then becomes the identification of a single consensus model describing the patterns common to all views [2]. In other cases significant discord may exist between the data in different views [3]. An effective data integration procedure must then reconcile these disagreements, identifying common patterns, while also preserving those that are unique to each view.

In this paper we propose a simple but effective algorithm for combining data from multiple views, based on a *late integration* strategy [4]. The proposed approach, referred to as Integration by Matrix Factorization (IMF), takes representative clusterings generated independently on each available view, constructs an intermediate matrix representation of those clusterings, and applies a factorization procedure to this representation to reconcile the groups arising from the individual views. The factorization procedure preserves the contribution of the

original clusters to the new groups, thereby highlighting the contribution made by each of the views. In addition we propose an entropy-based model selection procedure for automatically identifying the number of groups. To evaluate our approach we consider the problem of organizing topical news stories, represented by related text documents distributed across multiple views. These evaluations indicate that IMF can address common issues arising in real-world integration problems – such as disagreement between views, noisy views, and missing data.

This paper is organized as follows. Section 2 provides a brief overview of existing techniques for matrix factorization and fusing data from different sources. In Section 3 we discuss various issues that frequently arise when integrating multiple datasets in practice, and describe the proposed algorithm in detail. In Section 4 we present an empirical evaluation of the algorithm on synthetically-generated multi-view text datasets, followed by an evaluation on a real-world integration problem in Section 5. The paper finishes with some conclusions and suggestions for future work in Section 6.

2 Related Work

2.1 Matrix Factorization

Lee & Seung [5] proposed *Non-negative Matrix Factorization* (NMF), an unsupervised approach for dimensionality reduction, which approximates a data matrix as a product of factors that are constrained so that they will not contain negative values. By modeling each object as the additive combination of a set of non-negative basis vectors, a readily interpretable clustering of the data can be produced without further post-processing. These basis vectors are not required to be orthogonal, which facilitates the discovery of overlapping groups. The factorization process itself involves minimizing the difference between the original data and the approximation, most commonly by iteratively applying a pair of multiplicative update rules until the process converges to a local minimum [5].

2.2 Ensemble Clustering

In an unsupervised ensemble learning scenario we have access to a collection of "base clusterings", consisting of different clusterings generated on data originating from the same source. These clusterings represent the members of the ensemble. The primary aim of *ensemble clustering* [6] is to aggregate the information provided the ensemble members to produce a more accurate, stable clustering. A variety of strategies have been proposed to combine an ensemble to produce a single solution. For instance, the most widely-used strategy has been to consider information derived from the base clusterings to determine the level of *co-association* between each pair of objects in a dataset. Once a pairwise co-association matrix has been constructed, a standard algorithm such as single-linkage agglomerative clustering [7] or multi-level graph partitioning [6] is applied to produce a consensus clustering. The latter formulation was referred to by the authors as the Cluster-based Similarity Partitioning Algorithm (CSPA).

Rather than merely examining the pairwise relations between data objects, several authors have suggested examining the relations between the actual clusters contained in all base clusterings. Strehl & Ghosh [6] proposed the Hyper-Graph Partitioning Algorithm (HGPA), which involves transforming disjoint base clusterings to a hypergraph representation. Each node in the hypergraph represents a data object, and hyperedges are defined by the base cluster binary membership vectors. Subsequently a consensus clustering is produced by partitioning the hypergraph using the METIS algorithm [8].

The task of aggregating multiple clusterings can also be viewed as a *cluster correspondence* problem, where similar clusters from different base clusterings are matched together to produce a single "average clustering". Strehl & Ghosh [6] described a solution, referred to as the Meta-CLustering Algorithm (MCLA), which involves constructing a hypergraph where each hyperedge represents a cluster. The edges of the graph are then divided into a balanced k-way partition. Based on this edge partition, a majority voting scheme is used to assign data objects into the final clusters. The correspondence problem has been tackled by a number of other authors using *cumulative voting* ensemble clustering schemes, which are based on the assumption that there will be a direct relationship between individual clusters across all the base clusterings [9].

2.3 Data Integration

Blum & Mitchell [1] initially proposed the application of machine learning techniques in a *multi-view* setting, a problem which arises in domains where the data objects will naturally have several different representations. A useful broad distinction between techniques in this area was described by Pavlidis *et al.* [4], who identified three general data integration strategies: *early integration* involves the direct combination of data from several views into a single dataset before learning; *intermediate integration* involves computing separate similarity matrices on the views and producing a combined pairwise representation which is then passed to the learning algorithm; and *late integration* involves applying an algorithm to each individual view and subsequently combining the results.

While theoretical work in this area has largely focused on supervised learning problems, researchers have also considered the problem of producing clusterings from several different data sources. For instance, Bickel & Scheffer [2] proposed multi-view extensions of existing partitional and agglomerative clusterings algorithms. These algorithms were applied to the problem of clustering web pages, as represented by both textual information and hyperlinks. A general two-stage framework for reconciling discordant views in an unsupervised setting was described by Berthold & Patterson [3]. Other approaches have included minimizing the disagreement between views by casting the integration problem as an instance of bipartite spectral clustering [10], or as a semi-supervised clustering task where pairwise constraints generated from one view are used to influence the clustering process in another [11]. Both of these approaches are naturally limited to scenarios involving pairs of views.

3 Methods

3.1 Motivation

Given a set of views $\{V_1, \ldots, V_v\}$, let $\{x_1, \ldots, x_n\}$ denote the complete set of data objects present in the domain (*i.e.* $V_1 \cup V_2 \cdots \cup V_v$). The data integration task involves producing a complete clustering of the n objects to uncover all significant underlying "patterns" or groups present in the domain. In practice such integration tasks will often encounter one or more of the following issues:

Diversity of representation: In some views a feature-based representation will be available for data objects, while in other views only relation-based representations will be available, often in the form of graphs or networks.

Incomplete views: A representation for a data object in each view will not always be available. Rather, each view will often contain a subset of the total set of data objects in the domain.

Missing patterns: Patterns may be present in the data in one view, but largely or entirely absent from another view. As a consequence the number of patterns in each view will also vary.

Disagreement between views: The assignment of data objects to patterns may be inconsistent between views. Such disagreements can arise due to the unique characteristics of problem domain, or can simply be the result of noise within a view.

With the above requirements in mind, we propose a factorization-based formulation of the *late integration* [4] strategy for exploring domains were two or more related views exist. This approach, referred to as Integration by Matrix Factorization (IMF), takes representative clusterings generated independently on each individual view (using an algorithm appropriate for that view), constructs an intermediate representation of the clusterings, and decomposes this representation to reconcile the groups arising from the individual views. The fact that IMF operates on previously generated clusterings alone, rather than any specific representation of the original data, neatly avoids the diversity of representation issue. Late integration brings a number of additional benefits: the ability to harness parallel computing resources by processing large data views separately, the aggregation of information from views where privacy issues arise (*e.g.* financial, legal or commercially-sensitive data), and the facility to reuse knowledge available in existing legacy clusterings [6]. Later in Section 4 we demonstrate that the IMF algorithm can also address the other key integration issues of incomplete views, missing patterns, and disagreement between the set of available views.

3.2 Integration by Matrix Factorization

Intermediate representation. Formally we have access to a set of representative clusterings $\mathbb{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_v\}$, one per view, where $\mathcal{C}_h$ indicates the set of k_h clusters $\{c_h^1, \ldots, c_h^{k_h}\}$ generated on the view V_h. The sum $l = \sum_{i=1}^{v} k_i$ is the total number of clusters generated on all views. The clusterings may be generated by

an algorithm that produces a disjoint partition (*e.g.* standard k-means or the kernelized equivalent), probabilistic clusters (*e.g.* EM clustering), or arbitrary non-negative membership weights (*e.g.* NMF). Hierarchical clusterings can be combined by applying a suitable cut-off strategy to produce a disjoint partition. However, for the remainder of this paper we focus on disjoint clusterings.

The constituent clusterings in $\mathbb{C}$ can be represented by a set of non-negative membership matrices $\mathbb{M} = \{\mathbf{M}_1, \ldots, \mathbf{M}_v\}$, where $\mathbf{M_h} \in \mathbb{R}^{n \times k_h}$ represents the cluster membership of objects in $\mathcal{C}_h$ generated on view V_h. For objects which are not present or clustered in a given view, the corresponding row in the membership matrix of the clustering for that view will contain zero values. By transposing the matrices in $\mathbb{M}$ and stacking them vertically, we can construct a matrix of clusters $\mathbf{X} \in \mathbb{R}^{l \times n}$. Each row in $\mathbf{X}$ now corresponds to an individual cluster from the clusterings in $\mathbb{C}$, while each column corresponds to a data object in the original domain. Following the discussion in [6], conceptually we can view this representation as the adjacency matrix of a hypergraph consisting of n vertices and l weighted hyperedges. Alternatively we can interpret the columns of $\mathbf{X}$ as an embedding of the original objects in a new l-dimensional space.

Factorization process. The goal of the integration process is to project the clusters in $\mathbb{C}$ to a set of $k' < l$ new basis vectors or "meta-clusters", where k' represents the number of underlying patterns present in the domain. These meta-clusters represent the additive combinations of clusters generated on one or more different views. Clusters generated on the same view can also be grouped together. This may be desirable in cases where a pattern has been incorrectly split in a view, or where the constituent clusterings are generated at a higher resolution than is required for the integrated solution.

Formally, the process involves producing an approximation of $\mathbf{X}$ in the form of the product of two non-negative factors:

$$\mathbf{X} \approx \mathbf{PH} \quad \text{such that} \quad \mathbf{P} \geq 0 \, , \mathbf{H} \geq 0$$

where the rows of $\mathbf{P} \in \mathbb{R}^{l \times k'}$ represent the projection of the original clusters to a set of new basis vectors representing k' "meta-clusters". These meta-clusters can be additively combined using the coefficient values from the matrix $\mathbf{H} \in \mathbb{R}^{k' \times n}$ to reconstruct an approximation of the original set of clusters in $\mathbf{X}$. Furthermore, each column in $\mathbf{H}$ can be viewed as the membership of the original complete set of n data objects with respect to the k' meta-clusters.

To measure the reconstruction error between the original matrix $\mathbf{X}$ and the pair of factors $(\mathbf{P}, \mathbf{H})$ we can compute the Frobenius norm:

$$||\mathbf{X} - \mathbf{PH}||_{\mathsf{F}}^2 = \sum_{i=1}^{l} \sum_{j=1}^{n} [X_{ij} - (\mathbf{PH})_{ij}]^2 \tag{1}$$

To minimize Eqn. 1 we iteratively apply the multiplicative update rules proposed by Lee & Seung [5]:

$$P_{ic} \leftarrow P_{ic} \frac{(\mathbf{X}\mathbf{H}^{\mathsf{T}})_{ic}}{(\mathbf{P}\mathbf{H}\mathbf{H}^{\mathsf{T}})_{ic}} \qquad H_{cj} \leftarrow H_{cj} \frac{(\mathbf{P}^{\mathsf{T}}\mathbf{X})_{cj}}{(\mathbf{P}^{\mathsf{T}}\mathbf{P}\mathbf{H})_{cj}}$$

The rules are applied until the change in the objective (Eqn. 1) between one iteration and the next is below an arbitrarily small value. The computational cost of each iteration is $O(lnk')$ when using dense matrix multiplication operations.

The additive nature of the factorization procedure can be useful in interpreting the results of the integrating process. Based on the values in the projection matrix $\mathbf{P}$, we can calculate a matrix $\mathbf{T} \in \mathbb{R}^{v \times k'}$ indicating the contribution of the view V_h to each meta-cluster:

$$T_{hf} = \frac{\sum_{c_f^j \in \mathcal{C}_f} P_{jf}}{\sum_{g=1}^{l} P_{gf}} \tag{2}$$

That is, the sum of the projection weights in $\mathbf{P}$ for the clusters generated on V_h, normalized with respect to the total projection weight for each meta-cluster (*i.e.* the column sums of $\mathbf{P}$). A value T_{hf} close to 0 indicates that the view V_h has made little contribution to the f-th meta-cluster, while a value close to 1 indicates that the view V_h has made the predominant contribution.

To illustrate the integration process, Figure 1 shows a simple problem involving objects $\{x_1, \ldots, x_7\}$ represented in two views. The corresponding clusterings $\mathbb{C} = \{\mathcal{C}_1, \mathcal{C}_2\}$ are transformed to the intermediate representation $\mathbf{X}$, and factorization is applied to yield the matrices $(\mathbf{P}, \mathbf{H})$. The entries in $\mathbf{P}$ illustrate how the clusters from these clusterings are combined to produce $k' = 3$ meta-clusters. The actual object membership weights for these meta-clusters are shown in $\mathbf{H}$.

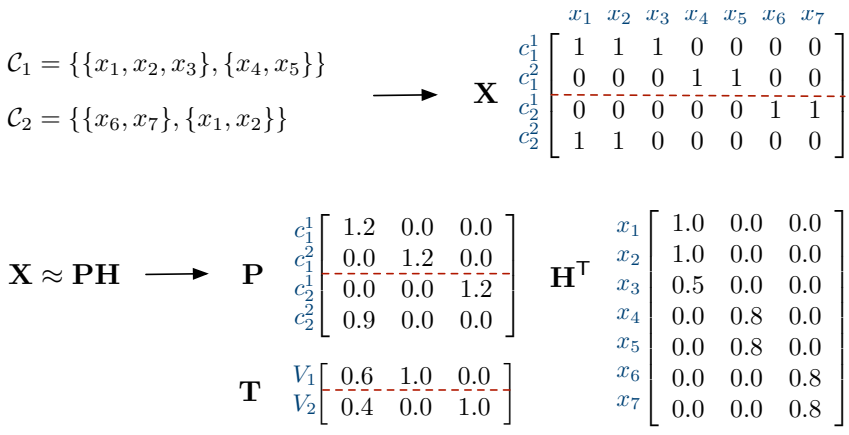

Fig. 1. Example of IMF applied to clusterings from two views generated in a domain containing 7 data objects. A value of $k' = 3$ is used for the number of meta-clusters.

The contributions made by the two views to the meta-clusters are given by the entries on the rows of the matrix $\mathbf{T}$.

Factorization initialization. The sensitivity of NMF-like algorithms to the choice of initial factors has been noted by a number of authors [12, 13]. While stochastic initialization is widely used in this context, ideally we would like to produce a single integrated clustering without requiring multiple runs of the integration process. Therefore to initialize the integration process, we populate the pair $(\mathbf{P}, \mathbf{H})$ by employing the deterministic NNDSVD strategy described by Boutsidis & Gallopoulos [13]. This strategy applies two sequential SVD processes to the matrix $\mathbf{X}$ to produce a pair of initial factors. In addition to being deterministic, NNDSVD is suitable in the context of integration as it has a tendency to produce comparatively sparse factors. As we shall see in the next section this is particularly desirable in the case of the projection matrix $\mathbf{P}$.

3.3 Model Selection

The selection of a suitable value for the number of meta-clusters k' is central to the data integration process. A value that is too low could force unrelated clusters to be grouped together, while a value that is too high could potentially cause the integration process to fail to merge related clusters from different views.

When selecting a model we consider the uncertainty of the mapping between clusters from different views, based on the uncertainty of the values in the projection matrix $\mathbf{P}$. Firstly we normalize the rows of $\mathbf{P}$ to unit length, yielding a normalized matrix $\hat{\mathbf{P}} \in [0, 1]$. In the ideal case each row in $\hat{\mathbf{P}}$ will contain a single value 1 and $(k' - 1)$ zeros, signifying that the corresponding base cluster has been perfectly matched to a single meta-cluster. This notion of uncertainty can be formally described in terms of the normalized entropy of the rows in $\hat{\mathbf{P}}$.

To illustrate this, we refer back to the previous example (Figure 1) of combining two clusterings. Figure 2 shows normalized project matrices corresponding to models for $k' = 3$ and $k' = 4$ respectively. In the latter case the integration procedure splits cluster c_1^1 between two meta-clusters (instead of matching it solely with c_2^2, which it subsumes as shown in Figure 1). Consequently the values in the first row of $\hat{\mathbf{P}}$ have a higher level of entropy. In contrast the matrix for $k' = 3$ shows a perfect match between each cluster from $\mathbb{C}$ and one of the three meta-clusters, suggesting that this model is more appropriate for the problem.

$$\hat{\mathbf{P}} \quad (k' = 3) \qquad \begin{matrix} c_1^1 \\ c_1^2 \\ c_2^1 \\ c_2^2 \end{matrix} \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \\ 1.0 & 0.0 & 0.0 \end{bmatrix} \qquad \hat{\mathbf{P}} \quad (k' = 4) \qquad \begin{matrix} c_1^1 \\ c_1^2 \\ c_2^1 \\ c_2^2 \end{matrix} \begin{bmatrix} 0.4 & 0.0 & 0.0 & 0.6 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

Fig. 2. Example of model selection for data integration applied to clusterings from two views, using two candidate values for the number of meta-clusters k'.

To identify an appropriate number of meta-clusters k', we can test values from a broad range $k \in [k_{min}, k_{max}]$ informed by the user's knowledge of the domain. For each candidate k we construct $\hat{\mathbf{P}}$. For each row j in this matrix we calculate the normalized entropy of the projection values:

$$e(\hat{\mathbf{P}}_j) = -\frac{1}{\log k} \sum_{h=1}^{k} P_{jh} \log (P_{jh}) \tag{3}$$

An evaluation $s(k) \in [0, 1]$ for the suitability of the model with k meta-clusters is given by subtracting the mean row entropy from 1:

$$s(k) = 1 - \frac{1}{l} \sum_{j=1}^{l} e(\hat{\mathbf{P}}_j) \tag{4}$$

A value for the final number of meta-clusters k' can be chosen so as to maximize the value of Eqn. 4.

In practice we observe that Eqn. 4 will not have an expected value of zero when combining randomly generated clusterings, and will exhibit a bias toward higher values of k. We can readily address this by employing the widely-used adjustment technique described in [14] to correct for chance agreement:

$$\hat{s}(k) = \frac{s(k) - \bar{s}(k)}{1 - \bar{s}(k)} \tag{5}$$

The value $\bar{s}(k)$ is the expected evaluation score for a factorization containing k meta-clusters. In practice we can find an approximation for $\bar{s}(k)$ by applying the following for a sufficiently large number of runs: take a given intermediate matrix $\mathbf{X}$, randomly permute the values in the columns, apply factorization with parameter k, and recalculate Eqn. 4. The expected value is given by the mean of $s(k)$ across all permutations.

3.4 Ensemble Multi-View Integration

The "one-shot" integration scenario described in Section 3.2 assumes the availability of a definitive clustering for each view. However, in many cases a variety of different clusterings may be available for each view – either generated on different subsets of the data in a view, produced using different parameter values, or simply as a result of the use of a clustering algorithm that converges to different local minima (*e.g.* k-means with random initialization). In many cases ensemble clustering techniques can harness the diversity present in such collections of clusterings [6]. We can naturally apply the IMF approach in a multi-view ensemble clustering setting, where $\mathbb{C}$ contains multiple clusterings generated on each view. As we shall see in our evaluation in Section 5, IMF can often take advantage of the diversity in a multi-view ensemble to produce a superior clustering.

4 Evaluation on Synthetic Multi-View Data

To evaluate the ability of the IMF approach described in Section 3 to handle a number of key issues that arise in data integration problems, we applied IMF to cluster multiple different views artificially produced from single-view news text corpora. Since we can assume that a single news article consists of one or more segments of text (*e.g.* one or more consecutive paragraphs), we can construct views containing related segments of text. The overall task then becomes the identification of an accurate clustering of the entire collection based on the segment information provided in the different views. Using this synthetically generated data, we can examine the effectiveness of the proposed approach in the context of the requirements detailed in Section 3.1.

4.1 Dataset Construction

We make use of the *bbc* and *bbcsport* news corpora[1] which have been previously used in document clustering tasks [12], and produce multiple views for documents based on related text segments. The original *bbc* corpus contains a total of 2225 documents with 5 annotated topic labels, while the original *bbcsport* corpus contains a total of 737 documents also with 5 annotated labels. From each corpus we constructed new synthetic datasets with 2-4 views as follows:

1. We split each raw document into segments. This was done by separating the documents into paragraphs, and merging sequences of consecutive paragraphs until 1-4 segments of text remained, such that each segment was at least 200 characters long. Each segment is logically associated with the original document from which it was obtained.
2. The segments for each document were randomly assigned to views, with the restriction that at most one segment from each document was assigned to the same view.
3. Standard stemming, stop-word removal and TF-IDF normalization procedures were separately applied to the segments in the individual views.

Details of the six resulting multi-view datasets[2] are provided in Table 1. To quantify algorithm performance, we calculate the *normalized mutual information* (NMI) [6] between clusterings and the set of annotated label information provided for the original corpora. These annotations are derived from the categories assigned to the original online news articles. Since NMI evaluates disjoint clusterings, we convert weighted membership matrices to disjoint clusterings by assigning each document to the cluster for which it has the highest weight. Note that in all cases NMI scores are calculated relative to the entire corpus, rather than relative to the subset present in any individual view.

[1] Both available from `http://mlg.ucd.ie/datasets/bbc.html`
[2] Available from `http://mlg.ucd.ie/datasets/segment.html`

Table 1. Details of the synthetic multi-view text datasets.

Datasets	View	Documents
bbc-seg2	1	2125
	2	2112
bbc-seg3	1	1828
	2	1832
	3	1845
bbc-seg4	1	1543
	2	1524
	3	1574
	4	1549

Collection	View	Documents
bbcsport-seg2	1	644
	2	637
bbcsport-seg3	1	519
	2	531
	3	513
bbcsport-seg4	1	400
	2	410
	3	437
	4	432

4.2 "One-Shot" Multi-View Integration

To examine the effectiveness of the IMF approach, we consider the scenario of combining a set of v clusterings, each coming from a different view. To provide a set of representative clusterings for our synthetic views, we apply spectral clustering followed by weighted kernel k-means as described in [15]. Since our focus here is not on the generation of these constituent clusterings, for convenience we set the number of clusters to the correct number of labeled classes for the associated corpora. The representative clusterings also provide a reasonable baseline comparison. When applying IMF itself, we select a value for the parameter k' using the procedure proposed in Section 3.3, using a candidate range $k' \in [4, 12]$.

Incomplete views. As indicated by the figures in Table 1, the synthetic view construction methodology will naturally result in cases where documents will be represented by segments in some but not all of the views (*i.e.* the views are not complete). Therefore we can directly examine the ability of the proposed algorithm to deal with this scenario. A summary of the results of the "one-shot" experiments on the six synthetic datasets is given in Table 2. The mean and standard deviation of the NMI scores for the constituent clusterings are listed for comparison. In all cases the application of IMF produced integration

Table 2. Accuracy (NMI) of the IMF approach on synthetic multi-view data, using one clustering per view.

Dataset	k'	Base	IMF
bbc-seg2	4	0.77 ± 0.05	0.80
bbc-seg3	5	0.71 ± 0.01	0.83
bbc-seg4	5	0.60 ± 0.01	0.83
bbcsport-seg2	4	0.74 ± 0.05	0.80
bbcsport-seg3	10	0.54 ± 0.05	0.62
bbcsport-seg4	6	0.39 ± 0.04	0.56

clusterings that were significantly better than those generated on the individual views.

Table 2 also shows the number of meta-clusters k' automatically selected by the entropy-based criterion (Eqn. 4). While the model selection procedure did not always exactly attain the "correct" number of clusters (both the *bbc* and *bbcsport* corpora contain 5 annotated topics), this value did appear in the top three recommended choices for five of the six datasets.

Missing patterns. To examine the behavior of IMF in scenarios were a pattern is entirely absent from a view, we took the synthetic datasets and removed all segments relating to a different randomly chosen label from each view (*i.e.* so that each view only contains segments pertaining to $k' - 1$ classes). We repeated this process for 20 runs, applying weighted kernel k-means on this data followed by IMF integration. For computational reasons, we use the same values of k' selected in the last set of experiments. Mean and standard deviation of NMI scores for these experiments are reported in Table 3. Again the IMF approach performs significantly better than the representative clusterings, and is successful in combining clusterings where an exact one-to-one mapping between the clusters in $\mathbb{C}$ does not necessarily exist. It is also worth noting that the NMI scores achieved are very close to those achieved when integrating clusterings generated on views with all patterns present (Table 2).

Disagreement between views. Next we examined the problem of discord between connected views. Specifically we considered the scenario where one view is considerably less informative than the others. In practice we selected one view at random and permuted 10% to 40% of the non-zero term values for each document, producing a noisy view on which a clustering was generated with spectral-initialized kernel k-means. IMF was then applied to integrate the noisy clustering together with the non-noisy clusterings from the $v - 1$ other views. In these experiments we set the value of k' to the "correct" number of class labels. We repeated the entire process for 30 runs and averaged the resulting NMI scores. Mean NMI scores for the base clusterings (both noisy and non-noisy) and the resulting integrated clusterings are given in Table 4.

Table 3. Mean accuracy (NMI) of the IMF approach on synthetic multi-view data *with missing patterns*, using one clustering per view.

Dataset	**Base**	**IMF**
bbc-seg2	0.76 ± 0.02	0.79 ± 0.03
bbc-seg3	0.66 ± 0.01	0.85 ± 0.03
bbc-seg4	0.56 ± 0.02	0.82 ± 0.04
bbcsport-seg2	0.69 ± 0.05	0.78 ± 0.03
bbcsport-seg3	0.51 ± 0.06	0.63 ± 0.04
bbcsport-seg4	0.39 ± 0.04	0.52 ± 0.04

Table 4. Mean accuracy (NMI) of the IMF approach on synthetic data using one clustering per view, where one of the views contains 10% to 40% noisy term values.

Dataset	**10% noise**		**20% noise**		**30% noise**		**40% noise**	
	Base	IMF	Base	IMF	Base	IMF	Base	IMF
bbc-seg2	0.79	0.84	0.79	0.83	0.75	0.80	0.72	0.77
bbc-seg3	0.69	0.83	0.67	0.81	0.64	0.78	0.59	0.75
bbc-seg4	0.57	0.81	0.56	0.79	0.54	0.78	0.49	0.72
bbcsport-seg2	0.71	0.81	0.66	0.74	0.65	0.74	0.60	0.66
bbcsport-seg3	0.53	0.65	0.51	0.63	0.47	0.58	0.41	0.47
bbcsport-seg4	0.40	0.53	0.38	0.51	0.35	0.49	0.26	0.35

A key test in this experiment is whether an integrated clustering can improve on its constituent clusterings in the presence of noisy views, rather than having performance equivalent to the "weakest link" among the views. As expected we observe that the meta-clusters produced by IMF remain significantly more accurate than the underlying constituent clusterings. Secondly, comparing the results to those in Table 2, we see that for 10-20% noise there is little decrease in clustering accuracy. For more extreme levels of noise, the IMF clustering on datasets derived from the *bbc* corpus remain reasonably accurate, while we see a greater effect on the datasets derived from the *bbcsport* corpus. In general these experiments suggest that the IMF approach is reasonably tolerant to disagreement between views, and cases where one view is weaker than the others.

5 Evaluation on Real-World Data

In this section we describe an evaluation of the proposed integration approach performed on a real-world multi-view document clustering task – namely that of clustering topical news stories where multiple reports of the same news story are available from different news sources. We constructed a new multi-view dataset[3], referred to as the *3sources* collection, from three well-known online news sources: BBC[4], Reuters[5], and The Guardian[6]. This dataset exhibits a number of common aspects of multi-view problems highlighted previously – notably that certain stories will not be reported by all three sources (*i.e.* incomplete views), and the related issue that sources vary in their coverage of certain topics (*i.e.* partially missing patterns).

In total we collected 948 news articles covering 416 distinct news stories from the period February–April 2009. Of these stories, 169 were reported in all three sources, 194 in two sources, and 53 appeared in a single news source. Each story was manually annotated with one or more of the six topical labels: *business*,

[3] Available from `http://mlg.ucd.ie/datasets/3sources.html`
[4] `http://news.bbc.co.uk`
[5] `http://reuters.co.uk`
[6] `http://www.guardian.co.uk`

Table 5. The distribution of dominant topic labels for stories in the *3sources* collection. The overall total number of stories per label is given, as well as the number of articles present within each individual view.

Label	Overall	BBC	Guardian	Reuters
business	122	87	78	94
entertainment	70	53	41	43
health	57	45	24	27
politics	61	48	40	23
sport	90	81	76	71
technology	67	38	43	36

entertainment, *health*, *politics*, *sport*, *technology*. These roughly correspond to the primary section headings used across the three news sources. To facilitate comparisons using the NMI measure, in our evaluation we consider only the dominant topic for each news story, yieldings a disjoint set of annotated classes as shown in Table 5.

5.1 "One-Shot" Multi-View Integration

For our first evaluation on the *3sources* data, we consider a "one-shot" integration process. Once again we generate a representative clustering on each view using weighted kernel k-means [15], setting the value of k to the number of labels. A value $k' = 7$ for the number of meta-clusters was automatically selected using the entropy criterion described in Section 3.3. It is interesting to note that the additional cluster reflects the fact the integration procedure identifies two distinct clusters for "business and finance" – one cluster largely pertaining to reports on the global economic downturn, the other cluster containing stories directly related to business and finance, but also containing stories from *sport* and *entertainment* that have a business or financial dimension. The presence of this latter group reflects the actual overlapping nature of the topics in the collection. However, as noted earlier we focus on the disjoint labels during our evaluation to allow comparison with algorithms producing disjoint clusters.

Table 6 shows a comparison of the performance of the proposed approach to the clusterings produced on documents from the individual news sources.

Table 6. Performance of IMF on the *3sources* collection, compared with clusterings generated on individual views.

Algorithm/View	NMI	Assigned
Weighted kernel k-means (*BBC*)	0.65	85%
Weighted kernel k-means (*The Guardian*)	0.52	73%
Weighted kernel k-means (*Reuters*)	0.55	71%
Integration by Matrix Factorization	0.71	100%

IMF out-performs the three weighted kernel clusterings, and the resulting integrated clustering is considerably more informative than those generated on the Guardian and Reuters views. We observed that including these "weaker" sources of information does not significantly impact upon the effectiveness of the data integration process.

5.2 Ensemble Multi-View Integration

In the second evaluation performed on this collection, we consider the multi-view ensemble clustering problem described in Section 3.4. To generate the individual ensemble members, we use standard k-means with random initialization and cosine similarity. The number of base clusters is set to the number of labels, and the parameter value $k' = 7$ determined in Section 5.1 is used for the number of meta-clusters. As well as examining the performance of IMF, for comparison we also applied several well-known ensemble clustering algorithms from the literature: the CSPA co-association clustering approach [6], the HGPA and MCLA hypergraph-based methods [6], and the correspondence clustering cumulative voting method described in [9]. In total we conducted 30 runs, each involving the generation and integration of 100 different base clusterings per view.

Table 7 summarizes the experimental results for the four approaches under consideration, in terms of average, minimum and maximum NMI scores achieved over the 30 runs. The range of NMI scores for the complete set of 9000 base clusterings is also given. On average 76% of the total number of news stories was assigned in each base clustering, reflecting the incomplete nature of the views. Even though the information provided by the base clustering was often of very poorly quality, we see that most of the integration algorithms performed reasonably well, with the exception of the HGPA procedure. On average the IMF approach was the most successful of the techniques under consideration, suggesting that it frequently availed of the information provided by the "weak" but diverse clusterings generated on the three views.

Table 7. Accuracy (NMI) of IMF on the *3sources* collection, compared with four well-known ensemble clustering algorithms. The NMI scores for the constituent base clusterings are also listed.

Algorithm	**Mean**	**Min**	**Max**
Base	0.37	0.02	0.66
IMF	**0.78**	0.70	0.80
CSPA	0.62	0.61	0.64
HGPA	0.47	0.36	0.60
MCLA	0.72	0.65	0.79
Voting	0.74	0.67	0.79

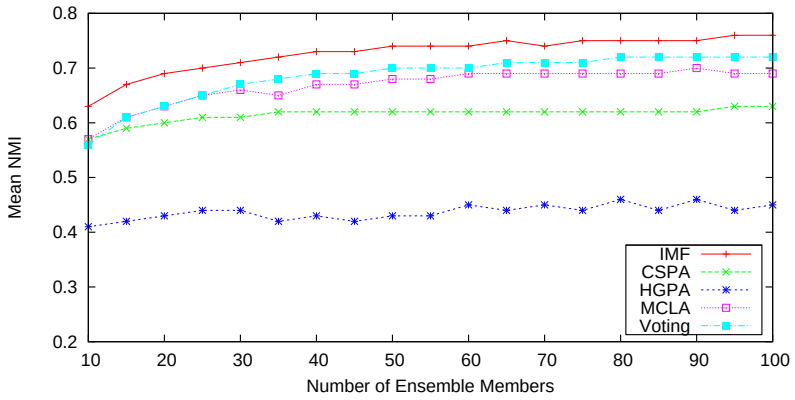

Fig. 3. Plot of mean clustering accuracy (NMI) scores for IMF compared to popular ensemble clustering algorithms, for ensembles of varying size generated on the *3sources* collection.

Effect of ensemble size. An important issue in ensemble clustering that is often neglected is the effect of ensemble size (*i.e.* the number of base clusterings in the ensemble) on clustering performance. For larger datasets, even with the availability of parallel computing resources, the number of clusterings that can reasonably be generated can often be strictly limited. As the size of the ensemble decreases, the ensemble multi-view clustering task approaches the one-shot integration task. However we still may face the problem of having access to only a set of weak or unrepresentative clusterings. Therefore it is desirable to employ an integration approach that will be effective when given a relatively small set of potentially weak base clusterings.

To examine this issue, we compare the behavior of the IMF algorithm with that of the four alternative approaches used above, as the number of ensemble members increases. Specifically we consider ensemble sizes $\in [5, 100]$ of k-means clusterings generated on the *3sources* collection, with an approximately equal number of clusterings per view. To account for variability in the results we repeat the process over 30 trials with different sets of base clusterings. As before, a value of $k' = 7$ was used as the number of final clusters. The results of the comparison are shown in Figure 3. We observe that IMF shows superior clustering accuracy in comparison to the alternative integration algorithms, particularly for smaller ensemble sizes – an NMI score of at least 0.70 was generally obtained after 25 clusterings have been added.

6 Conclusion

In this paper we presented a simple but effective approach for performing unsupervised data integration on two or more connected views. Experiments on synthetic and real-world multi-view text datasets yielded encouraging results,

both in tasks where a single representative clustering was available for each view, and in cases where a larger diverse collection of clusterings was available for integration. In the latter task the proposed IMF approach out-performed a number of popular ensemble clustering algorithms. An additional aspect of IMF is that the additive nature of the factorization process provides an aid in interpreting the output of the integration process.

Acknowledgments. This work is supported by Science Foundation Ireland Grant Nos. 05/IN.1/I24 and 08/SRC/I140 (Clique: Graph & Network Analysis Cluster)

References

1. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. Proc. 11th Annual Conference on Computational learning theory (1998) 92–100
2. Bickel, S., Scheffer, T.: Multi-view clustering. In: Proc. 4th IEEE International Conference on Data Mining. (2004) 19–26
3. Berthold, M., Patterson, D.: Towards learning in parallel universes. Proc. 2004 IEEE International Conference on Fuzzy Systems **1** (2004)
4. Pavlidis, P., Weston, J., Cai, J., Noble, W.: Learning Gene Functional Classifications from Multiple Data Types. Journal of Computational Biology **9**(2) (2002) 401–411
5. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401** (October 1999) 788–91
6. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. JMLR **3** (2002) 583–617
7. Jain, A.K., Fred, A.: Data clustering using evidence accumulation. In: Proc. 16th International Conference on Pattern Recognition. Volume 4. (2002) 276–280
8. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing **2096**(1) (1998) 359–392
9. Dimitriadou, E., Weingessel, A., Hornik, K.: A combination scheme for fuzzy clustering. International Journal of Pattern Recognition and Artificial Intelligence **16**(7) (2002) 901–912
10. de Sa, V.: Spectral clustering with two views. In: ICML Workshop on Learning With Multiple Views. (2005)
11. Zeng, E., Yang, C., Li, T., Narasimhan, G.: On the Effectiveness of Constraints Sets in Clustering Genes. Proc. 7th IEEE International Conference on Bioinformatics and Bioengineering (BIBE'07) (2007) 79–86
12. Greene, D., Cunningham, P.: Producing accurate interpretable clusters from high-dimensional data. In: Proc. 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'05). (2005) 486–494
13. Boutsidis, C., Gallopoulos, E.: SVD based initialization: A head start for non-negative matrix factorization. Pattern Recognition (2008)
14. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification (1985) 193–218
15. Dhillon, I.S., Guan, Y., Kulis, B.: Kernel k-means: spectral clustering and normalized cuts. In: Proc. 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2004) 551–556