Weighted Consensus Clustering

Tao Li[*] Chris Ding[†]

Abstract

Consensus clustering has emerged as an important extension of the classical clustering problem. We propose *weighted consensus clustering*, where each input clustering is weighted and the weights are determined in such a way that the final consensus clustering provides a better quality solution, in which clusters are better separated comparing to standard consensus clustering. Theoretically, we show that a reformulation of the well-known L_1 regularization LASSO problem is equivalent to the weight optimization of our weighted consensus clustering, and thus our approach provides sparse solutions which may resolve the difficult situation when the input clusterings diverge significantly. We also show that the weighted consensus clustering resolves the redundancy problem when many input clusterings correlate highly. Detailed algorithms are given. Experiments are carried out to demonstrate the effectiveness of the weighted consensus clustering.

1 Introduction

Consensus clustering has emerged as an important elaboration of the classical clustering problem. *Consensus clustering*, also called *aggregation of clustering (or partitions)*, refers to the situation in which a number of different (input) clusterings [1] have been obtained for a particular dataset and it is desired to find a single (consensus) clustering which is a better fit in some sense than the existing clusterings. Many additional problems can be reduced to the problem of consensus clustering; these include ensemble clustering, clustering of heterogeneous data sources, clustering with multiple criteria, distributed clustering, three-way clustering, and knowledge reuse [22][17].

Many approaches have been developed to solve consensus clustering problems over recently years [10][22][9][16][13]. There is, however, a significant drawback in current consensus clustering approaches, i.e., all input clustering solutions are treated equally, despite the facts that (1) different input clusterings could differ significantly, and (2) subsets of input clusterings could be highly correlated. Current approaches are essentially an averaging process and they are inadequate.

When different input clusterings differ significantly, the consensus by simply averaging is really a brute-force voting and there is no real "consensus" in their original meaning. The brute-force voting by highly divergent parties is not *stable*: the results could drift significantly when a few votes are missing or modified. On the other hand, when subsets of input clusterings are highly correlated, these redundant clusterings will tend to bias the final solution towards these correlated clusterings.

A further motivation is on the quality of the consensus. We wish to find a consensus clustering that is not only a consensus in the usual sense, but is also "better", i.e., in this consensus clustering, clusters are better separated, or equivalently, the clustering objective functions are improved.

With these motivations, in this paper, we propose the *weighted consensus clustering*. In our recent work [16], we show that consensus clustering problems can be formulated within Non-negative matrix factorization (NMF) framework [15, 5]. The weighted consensus clustering is based on the nonnegative matrix factorization framework. In this new approach, different input clusterings weigh differently, i.e., a weight for each input clustering is introduced, but the weights are automatically determined by an optimization process similar to a kernel matrix learning [14]. The central idea of this approach is to improve the quality of

[*] School of Computing and Information Sciences, Florida International University, 11200 SW 8th Street, Miami, FL 33199, USA

[†] Department of Computer Science and Engineering, University of Texas at Arlington, 416 Yates Street, Arlington, TX 76019, USA

[1] In this paper, we use clustering and partition interchangeably.

the consensus clustering, but we will show that our approach also overcomes the two drawbacks outlined earlier.

In particular, we will show (Section 4.1), when two input clustering P_1, P_2 are similar or highly correlated, the product of their weights w_1, w_2 will be small, i.e., either one weight is suppressed, or both weights are suppressed. This characteristics of our approach also overcomes, at least partially, the problem of redundant input clusterings.

As a theoretical contribution of this work, in Section 4.2, we show that the weight optimization part of our approach is equivalent to the well-known L_1 norm regression problem of LASSO [23], by a reformulation of the LASSO problem. This equivalence shows that our weighted consensus problem can be easily solved using existing efficient algorithms for LASSO. It also shows that the solutions for weights are sparse, i.e., only a subset of the input clusterings contribute to the final consensus clustering. This resolves the issues when input clusterings are highly divergent: only a small subset of those input clusterings (which are somehow close to the consensus) contribute to the final clustering, rather than a brute-force averaging.

Furthermore, the weights obtained in this approach could be useful for selecting input clusterings. Clearly, an input clustering with larger weight contributes more to the final consensus clustering. We will see in experiments that input clusterings which obtain large weights in the process are generally good clusterings as they tend to represent the more robust solutions with high diversity.

We wish to note that this weighted consensus clustering combines the recent trends in machine learning research. One recent trend is to use L_1 regularization to enforce a sparse solution and solution robustness. Many follow-up work along the direction of LASSO are sparse PCA extensions [26][2][27]. Another trend is kernel matrix learning [14], in which a number of input kernels (pairwise similarity matrices) are given and then the weights of a linear combination are learned through an optimization procedure. The weighted consensus clustering approach can be viewed as a combination of the kernel learning and LASSO approaches.

The rest of the paper is organized as follows. Section 2 introduces the notations and measures used for consensus clustering. Section 3 introduces the NMF-based formulation and algorithms for consensus clustering. Section 4 presents the weighted consensus clustering formulation. In addition, Section 4.1 discusses the characteristics of weighted consensus clustering for avoiding redundancy in input clusterings and Section 4.2 shows the equivalence between our approach with the L_1 norm regression of LASSO. Section 5 uses an illustrating example to demonstrate the advantages of the weighted consensus clustering. Section 6 presents the experimental results and Section 7 concludes.

2 Measures for Consensus Clustering

2.1 Notations Formally let $X = \{x_1, x_2, \cdots, x_n\}$ be a set of n data points. Suppose we are given a set of T clusterings (or partitions) $\mathcal{P} = \{P^1, P^2, \cdots, P^T\}$ of the data points in X. Each partition $P^t, t = 1, \cdots, T$, consists of a set of clusters $C^t = \{C_1^t, C_2^t, \cdots, C_k^t\}$ where k is the number of clusters for partition P^t and $X = \bigcup_{\ell=1}^{k} C_\ell^t$. Note that the number of clusters k could be different for different clusterings.

2.2 Connectivity Matrix There are several equivalent definitions of objective functions for aggregation of clustering. Following [10], we define the distance between two partitions P^1, P^2 as

$$d(P^1, P^2) = \sum_{i,j=1}^{n} d_{ij}(P^1, P^2),$$

where the element-wise distance is defined as

$$d_{ij}(P^1, P^2) = \begin{cases} 1 & (i,j) \in C_k(P^1) \text{ and } (i,j) \notin C_k(P^2) \\ 1 & (i,j) \in C_k(P^2) \text{ and } (i,j) \notin C_k(P^1) \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

where $(i,j) \in C_k(P^1)$ means that i and j belong to the same cluster in partition P^1 and $(i,j) \notin C_k(P^1)$ means that i and j belong to different clusters in partition P^1.

A simpler approach is to define the similarity between two clusterings (partitions). We define *connectivity matrix* as

$$M_{ij}(P^t) = \begin{cases} 1 & (i,j) \in C_k(P^t) \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

i.e., if x_i, x_j belong to the same cluster C_k, the connectivity between i, j is 1. We can easily see that

$$\begin{aligned} d_{ij}(P^1, P^2) &= |M_{ij}(P^1) - M_{ij}(P^2)| \\ &= [M_{ij}(P^1) - M_{ij}(P^2)]^2 \end{aligned}$$

because $|M_{ij}(P^1) - M_{ij}(P^2)| = 0$ or 1.

3 NMF-Based Formulation of Consensus Clustering

The weighted consensus clustering is based on the nonnegative matrix factorization(NMF) framework we recently proposed in [16]. In this section, we briefly introduce the NMF-Based formulation of consensus clustering.

We look for a consensus partition (consensus clustering) P^* which is the closest to all the given partitions:

$$\begin{aligned}\min_{P^*} J &= \frac{1}{T}\sum_{t=1}^{T} d(P^t, P^*) \\ &= \frac{1}{T}\sum_{t=1}^{T}\sum_{i,j=1}^{n} [M_{ij}(P^t) - M_{ij}(P^*)]^2 \\ &= \frac{1}{T}\sum_{t=1}^{T} ||M(P^t) - M(P^*)||_F^2.\end{aligned}$$

Let $U_{ij} = M_{ij}(P^*)$ denote the solution to this optimization problem. U is a connectivity matrix. Let the consensus (average) association between i and j be

$$\widetilde{M}_{ij} = \frac{1}{T}\sum_{t=1}^{T} M_{ij}(P^t). \tag{3.3}$$

Define the average squared difference from the consensus association $\widetilde{M}$:

$$\Delta M^2 = \frac{1}{T}\sum_t\sum_{ij}[M_{ij}(P^t) - \widetilde{M}_{ij}]^2.$$

Clearly, the smaller ΔM^2, the closer to each other the partitions are. This quantity is a constant. We have

$$\begin{aligned}J &= \frac{1}{T}\sum_t\sum_{ij}(M_{ij}(P^t) - \widetilde{M}_{ij} + \widetilde{M}_{ij} - U_{ij})^2 \\ &= \Delta M^2 + \sum_{i,j}(\widetilde{M}_{ij} - U_{ij})^2.\end{aligned}$$

Therefore consensus clustering takes the form of the following optimization problem:

$$\min_U \sum_{i,j=1}^{n} (\widetilde{M}_{ij} - U_{ij})^2 = ||\widetilde{M} - U||^2.$$

where the matrix norm is the Frobenius norm. Therefore Consensus clustering is equivalent to clustering the consensus association.

3.1 NMF Formulation Let U denote a solution of the consensus clustering problem. Being a connectivity matrix, U is characterized by a set of constraints. For example, Suppose i, j belong to the same cluster: $U_{ij} = 1$. If j and k belong to the same cluster ($U_{jk} = 1$), then i and k must belong to the same cluster ($U_{ik} = 1$). On the other hand, if j and k belong to a different cluster ($U_{jk} = 0$), then i and k must belong to a different cluster ($U_{ik} = 0$).

There are on the order of n^3 of these constraints. It was shown in [16] that these constraints can be enforced by clustering indicators. The clustering solution can be specified by clustering indicators $H = \{0,1\}^{n\times k}$, with the constraint that in each row of H there can only have one "1" and the rest must be zeros: $\sum_{\ell=1}^{k} H_{i\ell} = 1$.

Now it is easy to show that

$$U = HH^T, \quad \text{or} \quad U_{ij} = (HH^T)_{ij}. \tag{3.4}$$

First, we note $(HH^T)_{ij}$ is equal to the inner product between row i of H and row j of H. Second, we consider two cases. (a) When i and j belong to the same cluster, then row i must be identical to row j; in this case $(HH^T)_{ij} = 1$. (b) When i and j belong to different clusters, the inner product between row i and row j is zero.

With $U = HH^T$, the consensus clustering problem becomes

$$\min_H ||\widetilde{M} - HH^T||^2 \tag{3.5}$$

where H is restricted to an indicator matrix.

Now, let us consider the relaxation of the above integer optimization. The constraint that in each row of H there is only one nonzero element can be expressed as $(H^TH)_{k\ell} = 0$ for $k \neq \ell$. Also $(H^TH)_{kk} = |C_k| = n_k$. Let

$$D = \text{diag}(H^TH) = \text{diag}(n_1, \cdots, n_k).$$

We have $H^TH = D$. Now, we can write the optimization problem as

$$\min_{H^TH=D, H\geq 0} ||\widetilde{M} - HH^T||^2 \tag{3.6}$$

where H is relaxed into a continuous domain.

The optimization in Eq. (3.6) is easier to solve than the optimization of Eq. (3.5). However, in Eq. (3.6) we need to pre-specify D (the cluster sizes). But until the problem is solved we do not know D. Therefore we need to eliminate D. For this purpose, we define

$$\widetilde{H} = H(H^T H)^{-1/2},$$

Thus

$$HH^T = \widetilde{H}D\widetilde{H}^T,\ \widetilde{H}^T\widetilde{H} = H(H^T H)^{-1}H = I.$$

Therefore, the consensus clustering becomes the optimization

$$\text{(3.7)} \quad \min_{\widetilde{H}^T\widetilde{H}=I,\ \widetilde{H},D\geq 0} ||\widetilde{M} - \widetilde{H}D\widetilde{H}^T||^2, \text{ s.t. } D \text{ diagonal.}$$

Now both $\widetilde{H}^T$ and D are obtained together as the solution of the optimization problem. We do not need to pre-specify the cluster sizes.

3.2 Algorithms for NMF-based Consensus Clustering

We have shown that the consensus clustering problem is equivalent to a symmetric nonnegative matrix factorization problem. Generally, the consensus clustering problem of Eq. (3.7) can be solved by reducing it to the following problem:

$$\text{(3.8)} \quad \min_{Q\geq 0, S\geq 0} ||W - QSQ^T||^2,\ \ s.t.\ Q^T Q = I.$$

In Eq. (3.7) D is constrained to be a nonnegative diagonal matrix. More generally, we can relax D to be a generic symmetric nonnegative matrix.

The optimization problem in Eq. (3.8) can be solved using the following multiplicative update procedure:

$$\text{(3.9)} \quad Q_{jk} \leftarrow Q_{jk}\sqrt{\frac{(WQS)_{jk}}{(QQ^T WQS)_{jk}}},$$

$$\text{(3.10)} \quad S_{k\ell} \leftarrow S_{k\ell}\sqrt{\frac{(Q^T WQ)_{k\ell}}{(Q^T QSQ^T Q)_{k\ell}}}.$$

4 Weighted Consensus Clustering

Now we present the main result of this paper. The weighted consensus clustering is based on the nonnegative matrix factorization of approach of Section 3.2 and Eq.(3.5). The key idea is that when building the aggregate connectivity matrix, instead of using the simple average as in Eq.(3.3), we introduce the weights

$$\mathbf{w} = (w_1, w_2, \cdots, w_T)^T, \quad w_i \geq 0, \quad ||w||_1 = \sum_{i=1}^{T} w_i = 1.$$

and form the weighted aggregate connectivity matrix,

$$\text{(4.11)} \quad \widetilde{M} = \sum_{i=1}^{T} w_i M(P^t), \quad \text{or} \quad \widetilde{M}_{ij} = \sum_{i=1}^{T} w_i M_{ij}(P^t),$$

We define the weighted consensus clustering problem as

$$\text{(4.12)} \quad \min_{w,\ H} ||\widetilde{M} - \widetilde{H}\widetilde{H}^T||^2,$$

where the constraints on $w, \widetilde{H}$ are enforced. Recall that this NMF formulation or the original formulation of Eq.(3.5). are motivated by finding the consensus clustering for which its connectivity matrix $M(P^*)$ is closest to the input connectivity matrix $M_t(P)$ see Eq.(3.3).

We show that this weighted formulation also improves the quality of the consensus clustering. For this, we can re-write

$$||\widetilde{M} - \widetilde{H}\widetilde{H}^T||^2 = \mathrm{Tr}(\widetilde{M}\widetilde{M} - 2\widetilde{H}^T\widetilde{M}\widetilde{H} + \widetilde{H}\widetilde{H}^T\widetilde{H}\widetilde{H}^T)$$

Since the first and third terms are constant, we obtain a reformulation of the weighted consensus clustering problem

$$\text{(4.13)} \quad \max_{w,\ H} \mathrm{Tr}[\widetilde{H}^T\widetilde{M}\widetilde{H}]$$

In [25][4], it has been shown that the popular K-means clustering can be formulated in an identical way. Therefore, improving this objection function is equivalent to improve the quality of the clustering solution.

The optimization problem can be solved by iterating the following two steps:

Step 1: Solve for $\widetilde{H}$ while fixing $\mathbf{w}$. This can be done using the method described in Section 3.

Step 2: Solve for $\mathbf{w}$ while fixing $\widetilde{H}$. Note that

$$\begin{aligned} J &= Tr[(\widetilde{M} - \widetilde{H}\widetilde{H}^T)(\widetilde{M} - \widetilde{H}\widetilde{H}^T)] \\ &= Tr(\widetilde{M}^2 - 2\widetilde{H}^T\widetilde{M}\widetilde{H} + \widetilde{H}^T\widetilde{H}\widetilde{H}^T\widetilde{H}) \end{aligned}$$

Note that

$$Tr(\widetilde{H}^T\widetilde{M}H) = b^T\mathbf{w} \text{ where } b_i = \widetilde{H}^T M(P^i)\widetilde{H}$$

and

$$\begin{aligned} \widetilde{M}^2 &= w_1^2 M(P^1)^2 + \cdots + w_T^2 M(P^T)^2 \\ &\quad +2w_1w_2M(P^1)M(P^2) + \cdots \\ &= \mathbf{w}^T A\mathbf{w} \end{aligned}$$

where

$$A_{ij} = Tr[M(P^i)M(P^j)] = \sum_{uv} M(P^i)_{uv} M(P^j)_{uv}$$

measures the similarity between the two clusterings $M(P^i)$ and $M(P^j)$. Clearly, A is a semi-definite-positive matrix, a nice property which makes the problem tractable.

Thus for fixing $\widetilde{H}$, the optimization problem becomes

(4.14)

$$\min J = \mathbf{w}^T A\mathbf{w} - 2b^T\mathbf{w} + \text{const.} \quad \text{s.t.} \sum_{i=1}^{T} w_i = 1, w_i \geq 0$$

This is quadratic function optimization problem with linear constraints with T variables. Because A is s.d.p, and the feasibility region is convex, Thus Eq.(4.14) is a convex optimization problem and we can obtain the global solution. This can be solved using a standard quadratic programming code. Using quadratic programming for solving the weights in Eq.(4.14) is on a problem of just about hundreds of variables (i.e., weights for each input clustering) and thus can be computed quickly.

4.1 Reducing Redundancy in Clustering Ensemble

When collecting a large number of clusterings, quite often many clusterings could be close (similarly) to each other. These would easily skew the final consensus clustering. This is one of the motivation of this work.

Formulation Eq. 4.14 reveals an important property of the weighted consensus clustering problem — it automatically reduces the redundancy.

Since the objective function is generally dominated by the quadratic term, $\mathbf{w}^T A\mathbf{w}$, we may consider the first order approximation by considering

$$\min_{w} \mathbf{w}^T A\mathbf{w} \quad \text{s.t.} \sum_{i=1}^{T} w_i = 1, w_i \geq 0$$

Because we minimize $\mathbf{w}^T A\mathbf{w}$, it reasonable to expect that a large A_{ij} will lead to a small $w_i w_j$. i.e., either one of w_i, w_j will be small, or both of w_i, w_j will be small. In other words, if two clusterings are similar, the corresponding weights in the final solution will tend to be small.

4.2 Relation to L_1-norm and Sparsity via LASSO

In this section, we establish the equivalence of the weight optimization part of the weighted consensus clustering to LASSO (sign-preserved LASSO to be precise, see later). From this, we infer that the final solution to the weighted consensus clustering must be sparse.

Let

$$y = (y_1, y_2, \cdots, y_n)^T \in \Re^n,$$

be n experimental measurements, and

$$X = (x_1, x_2, \cdots, x_n) \in \Re^{p\times n}$$

be n p-dimensional data points. Let

$$\beta = (\beta_1, \beta_2, \cdots, \beta_p)^T \in \Re^p$$

be the regression parameters to be determined. Standard linear regression minimizes

$$\begin{aligned} \sum_{i=1}^{n} (y_i - \beta^T x_i)^2 &= ||y^t - \beta^T X||^2 \\ &= \beta^T X X^T \beta - 2(Xy)^T \beta + y^T y \end{aligned}$$

which gives the usual ”unconstrained” solution

$$\bar{\beta} = (XX^T)^{-1} Xy$$

LASSO solves the L_1 norm constrained problem

$$\sum_{i=1}^{n} (y_i - \beta^T x_i)^2 \quad s.t. \quad ||\beta||_1 \leq t \leq ||\bar{\beta}||_1$$

Now we reformulate the LASSO using nonnegative weights in order to make a connection to our weighted consensus clustering.

We set

$$\beta = [\hat{\beta}_1 \text{sign}(\bar{\beta}_1), \hat{\beta}_2 \text{sign}(\bar{\beta}_2), \cdots, \hat{\beta}_p \text{sign}(\bar{\beta}_p)]^T$$

Let $\hat{X} = \text{diag}(\text{sign}(\bar{\beta}_1), \cdots, \text{sign}(\bar{\beta}_p))X$ to absorb the signs, the LASSO problem can be reformulated into

$$\min_{\hat{\beta}} \; \hat{\beta}^T \hat{X}\hat{X}^T \hat{\beta} - 2(\hat{X}y)^T \hat{\beta} + y^T y \quad s.t. \quad ||\hat{\beta}||_1 \leq t$$

Now, we note two characteristics of LASSO.

1. Active constraint. In LASSO, the optimal solution always happens at active constraint, i.e., the solution satisfies $||\beta||_1 = t$. Thus we can replace $||\beta||_1 \leq t$ by $||\beta||_1 = t$.

2. Sign preservation. In LASSO, in most cases, the signs of the final solution of β (each component) are the same as the unconstrained solution $\hat{\beta}$. Thus we can replace the constraint $||\hat{\beta}||_1 = t$ as $\sum_{j=1}^{p} \hat{\beta}_j = t$. In the rare case, this sign preservation does not hold.

With these two observations, we can write the sign-preserved LASSO as

$$\min_{\hat{\beta}} \; \hat{\beta}^T \hat{X}\hat{X}^T \hat{\beta} - 2(\hat{X}y)^T \hat{\beta} + y^T y \quad s.t. \quad \sum_{j=1}^{p} \hat{\beta}_j = t.$$

Therefore, this sign-preserved LASSO is identical to the weight optimization part of our weighted consensus clustering. Furthermore, the meanings of quadratic terms in two problems are also the same. Here the matrix coefficients of quadratic term $(\hat{X}\hat{X}^T)_{ij}$ are the covariances of x_i and x_j, which are similar to A_{ij}'s in our consensus clustering where A_{ij} measures the similarity between partitions P_i and P_j. The difference between our weighted consensus clustering and LASSO is that we fix the sum of weights to be 1 (i.e., a natural normalization), where in LASSO, t controls the amount of shrinkage, which is not 1. But this is just difference in scale (we can rescale A, b) and the natures of the solutions are the same.

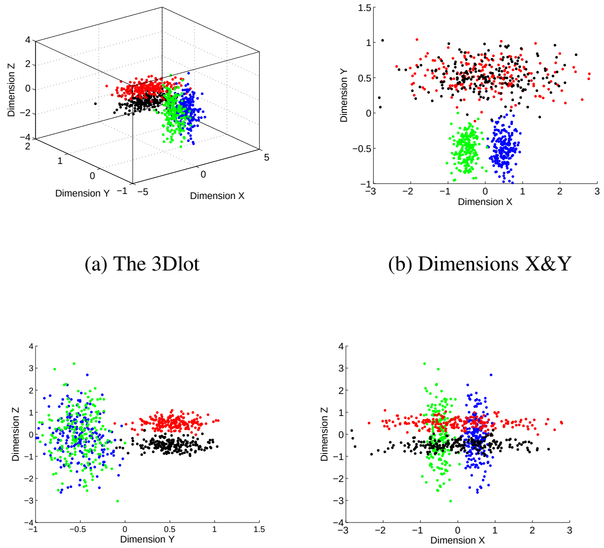
It is well-known that LASSO solutions are sparse, i.e., many components in β are zero. From the above equivalence, we expect that the solution of weights in our weighted consensus clustering are sparse. Or equivalently, we can expect many small weights, especially on those corresponding the redundant clusterings.

5 An Illustrative Example

To illustrate weighted consensus clustering, we create a dataset with 400 instances in three dimensions. The dataset is divided into four clusters of 100 instances, each existing in only two of the three dimensions [21]. The data are plotted in Figure 1, where the four clusters are indicated by different colors (or gray scales).

- The first cluster is generated from a normal distribution with mean $(0.5; -0.5; 0)$ and standard deviation $(0.2; 0.2; 1)$;

- The second cluster is generated from a normal distribution with mean $(-0.5; -0.5; 0)$ and standard deviation $(0.2; 0.2; 1)$;



(a) The 3Dlot

(b) Dimensions X&Y

(c) Dimensions Y&Z

(d) Dimensions X&Z

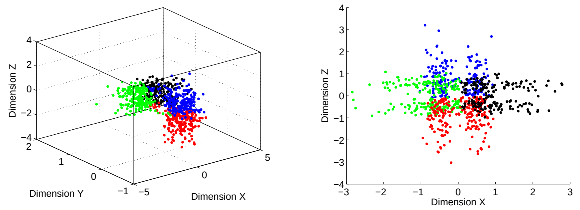
Figure 1: The data set. (a) A 3D plot of the dataset. (b)-(d) The data plotted in each set of two dimensions.

- The third cluster is generated from a normal distribution with mean $(0; 0.5; 0.5)$ and standard deviation $(1; 0.2; 0.2)$;

- The fourth cluster is generated from a normal distribution with mean $(0; 0.5; -0.5)$ and standard deviation $(1; 0.2; 0.2)$.

In Figure 1(b) and Figure 1(c), we see that only two clusters are clearly visible while the remaining two clusters overlap. In Figure 1(d), the four clusters are more visible, however they are still mixed together and not properly separated. K-means clustering can not separate these four clusters. It should also be noted that feature transformation techniques such as Principal Component Analysis (PCA) do not work well in this case [21].

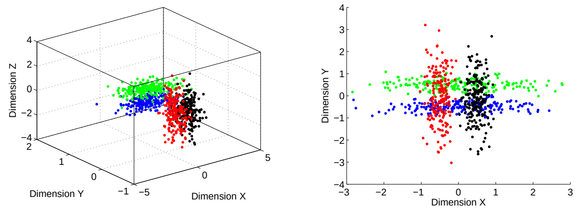
For this dataset, the average clustering accuracy (clustering accuracy is defined in Section 6) of K-means on original dataset over 30 trials is 0.50 (see Figure 2). Now we compute the consensus matrix by applying K-means algorithm (with K=20) to the original dataset 20 times, the NMF-based consensus clustering leads to a much improved clustering accuracy of 0.95. Weighted consensus further improves the clustering accuracy to



(a) The results of K-means clustering

(b) The results of K-means plotted in the first and third dimensions

Figure 2: Results of K-means clustering. Different clusters obtained by Kmeans clustering are indicated using different colors (or gray scales).



(a) The results of weighted consensus clustering

(b) The results of weighted consensus clustering plotted in the first and third dimensions

Figure 3: Results from weighted consensus clustering. Different clusters are indicated using different colors (or gray scales).

Measures	Top Five Clusterings	All Clusterings
AA	0.74	0.65
D1	0.35	0.32
D2	0.17	0.15

Table 1: Comparison of weighted consensus clustering and NMF-based consensus clustering on the example dataset. AA represents the average clustering accuracy of the related clusterings. $D1$ and $D2$ are two diversity measures defined in Section 6.2.

0.985 and its results are shown in Figure 3.

A new and useful feature of the weighted consensus clustering is that the weights obtained in weighted consensus clustering can be used to select individual input clusterings, i.e, assess how important of each input clustering. Table 1 shows the average clustering accuracy and diversity (clustering diversity is explained and defined in Section 6) of the top five selected clusterings based on the obtained weights. It is shown that the partitions associated with high weights are usually with high quality and diversity.

6 Experiments

In this section, we conduct experiments on real world datasets to evaluate the effectiveness of our weighted consensus clustering algorithm.

6.1 Improving Clustering Results The goal of this set of experiments is to evaluate the extent to which weighted consensus clustering can improve the robustness of traditional clustering algorithms. We compare our weighted consensus clustering with the results of running K-means on the original dataset, and the results of running K-means on the consensus similarity matrix. The consensus similarity matrix is obtained by running K-means 30 times. We also compare our weighted consensus clustering with the NMF-based consensus clustering [16], the cluster-based similarity partitioning algorithm (CSPA), and the HyperGraph Partitioning Algorithm (HGPA) described in [22].

6.1.1 Dataset Description We conduct experiments using the datasets summarized in Table 2. The number of classes ranged from 2 to 20, the number of samples ranged from 47 to 2900, and the number of dimensions

Datasets	# Samples	# Dimensions	# Class
CSTR	475	1000	4
Digits389	456	16	3
Glass	214	9	7
Ionosphere	351	34	2
Iris	150	4	3
Protein	116	20	6
LetterIJL	227	16	3
Log	1367	200	8
Reuters	2900	1000	10
Soybean	47	35	4
WebACE	2340	1000	20
WebKB4	4199	1000	4
Wine	178	13	3
Zoo	101	18	7

Table 2: Descriptions of datasets.

ranged from 4 to 1000. Further details are as follows:

- Nine datasets (Digits389, Glass, Ionosphere, Iris, LetterIJL, Protein, Soybean, Wine, and Zoo) are from UCI data repository [7]. Digits389 is a randomly sampled subset of three classes: $\{3,8,9\}$ from digits dataset. LetterIJL is a randomly sampled subset of three $\{I,J,L\}$ from Letters dataset.

- Five datasets (CSTR, Log, Reuters, WebACE, WebKB4) are standard text datasets that are often used as benchmarks for document clustering. The documents are represented as the term vectors using vector space model. These document datasets are pre-processed (removing the stop words and unnecessary tags and headers) using the rainbow package [19].

- CSTR is the dataset of the abstracts of technical reports published in Computer Science departments between 1991 and 2002. The dataset contains 476 abstracts, which are divided into four research areas: Natural Language Processing(NLP), Robotics/Vision, Systems, and Theory.

- The Log dataset contains 1367 text messages of system log from different desktop machines describing the status of computer components. These messages are divided into 8 different situations.

- The Reuters dataset is a subset of the Reuters-21578 Text Categorization Test collection containing the 10 most frequent categories among the 135 topics.

- The WebACE dataset contains 2340 documents consisting of news articles from 20 different topics in October 1997 collected in WebACE project [12].

- The WebKB dataset contains webpages gathered from university computer science departments. There are about 8280 documents and they are divided into 7 categories: student, faculty, staff, course, project, department and other. The WebKB4 dataset is the subset of WebKB associating with four most populous entity-representing categories, i.e., student, faculty, course and project.

6.1.2 Experimental Results All the above datasets come with labels. Viewing these labels as indicative of a reasonable clustering, we define the following accuracy measure [18][6]:

$$Accuracy = \max\Big(\sum_{C_k,L_m} T(C_k,L_m)\Big)/n, \tag{6.15}$$

where n is the number of data points, C_k denotes the k-th cluster, L_m is the m-th class and $T(C_k,L_m)$ is the number of data points that belong to class m are assigned to cluster k. Accuracy is thus computed as the maximum sum of $T(C_k,L_m)$ for all pairs of clusters and classes, and these pairs have no overlap.

From Table 3, we observe that Weighted consensus clustering and NMF-based consensus clustering improve K-means clustering on all datasets expect Reuters. Weighted consensus clustering achieves better performance than NMF-based algorithm on 6 datasets and their performance on the remaining datasets are very close. In summary, the experiments clearly demonstrate the effectiveness of weighted consensus clustering for improving clustering performance and robustness.

6.2 Measuring Consensus Diversity Diversity within consensus clustering is an important factor affecting the performance. An ensemble formed from identical base partitions would not outperform its individual members and the combination of multiple

	K-means	KC	CSPA	HPGA	NMFC	WC
CSTR	0.45	0.37	0.50	0.62	0.56	0.64
Digits389	0.59	0.63	0.78	0.38	0.73	0.71
Glass	0.38	0.45	0.43	0.40	0.49	0.49
Ionosphere	0.70	0.71	0.68	0.52	0.71	0.71
Iris	0.83	0.72	0.86	0.69	0.89	0.89
Protein	0.53	0.59	0.59	0.60	0.63	0.65
Log	0.61	0.77	0.47	0.43	0.71	0.69
LetterIJL	0.49	0.48	0.48	0.53	0.52	0.52
Reuters	0.45	0.44	0.43	0.44	0.43	0.44
Soybean	0.72	0.82	0.70	0.81	0.89	0.91
WebACE	0.41	0.35	0.40	0.42	0.48	0.46
WebKB4	0.60	0.56	0.61	0.62	0.64	0.63
Wine	0.68	0.68	0.69	0.52	0.70	0.72
Zoo	0.61	0.59	0.56	0.58	0.62	0.70

Table 3: Results on consensus clustering, shown is clustering accuracy . The results are obtained by averaging over five trials. KC represents the results of applying K-means to a consensus similarity matrix, NMFC represents the NMF-based consensus clustering and WC represents the weighted consensus clustering.

clusterings is useful only if there is disagreement among them [11]. To get a better understanding on the behaviors of the weighted consensus clustering, we take a closer look at the diversity of the cluster ensembles.

Let $ARI(P^i,P^j)$ and $NMI((P^i,P^j)$ denote the adjusted Rand index [20] and normalized mutual information [1] between two partitions P^i and P^j. We use following two approaches for measuring the diversity of a collection of T partitions:

$$D_1 = \frac{2}{T(T-1)} \sum_{i=1}^{T-1} \sum_{j=t+1}^{T} (1 - ARI(P^i,P^j)). \tag{6.16}$$

$$D_2 = \frac{2}{T(T-1)} \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} (1 - NMI(P^i,P^j)). \tag{6.17}$$

Note that D_1 and D_2 measure the pair-wise ensemble similarity using adjusted rand Index and normalized mutual information respectively. The larger the measures, the more diverse the ensembles are. In Figure 4, we compare the average clustering accuracy and diversity of the top five selected clusterings based on the obtained weights with those of all the input clusterings on several datasets. We can observe that on 4 datasets (CSTR, Wine, Iris and Soybean), the top 5 selected clusterings tend to be more diverse and accurate. On Glass and Zoo datasets, the selected clusterings are less diverse. It should be noted that the relationships of diversity and accuracy might not be monotonic. In some cases, ensembles exhibiting a moderate level of diversity produce more accurate clusterings [3]. The reason might be that the class labels might not necessarily correspond to natural clusters. In summary, as we discussed in Section 5, the weights obtained in weighted consensus clustering can be used to select individual input clusterings, i.e, assess how important of each input clustering.

6.3 Clustering Heterogeneous Data

The goal of this set of experiments is to investigate the effectiveness of NMF-based clustering aggregation for the clustering of heterogeneous data. In particular, we study the problem of identifying the artist style; i.e., we cluster songs into groups denoted by the artists using both content and lyrics [18]. Ellis et al. [8] point out that similarity between artists reflects personal tastes and suggest that different measures have to be combined so as to achieve reasonable results in this problem.

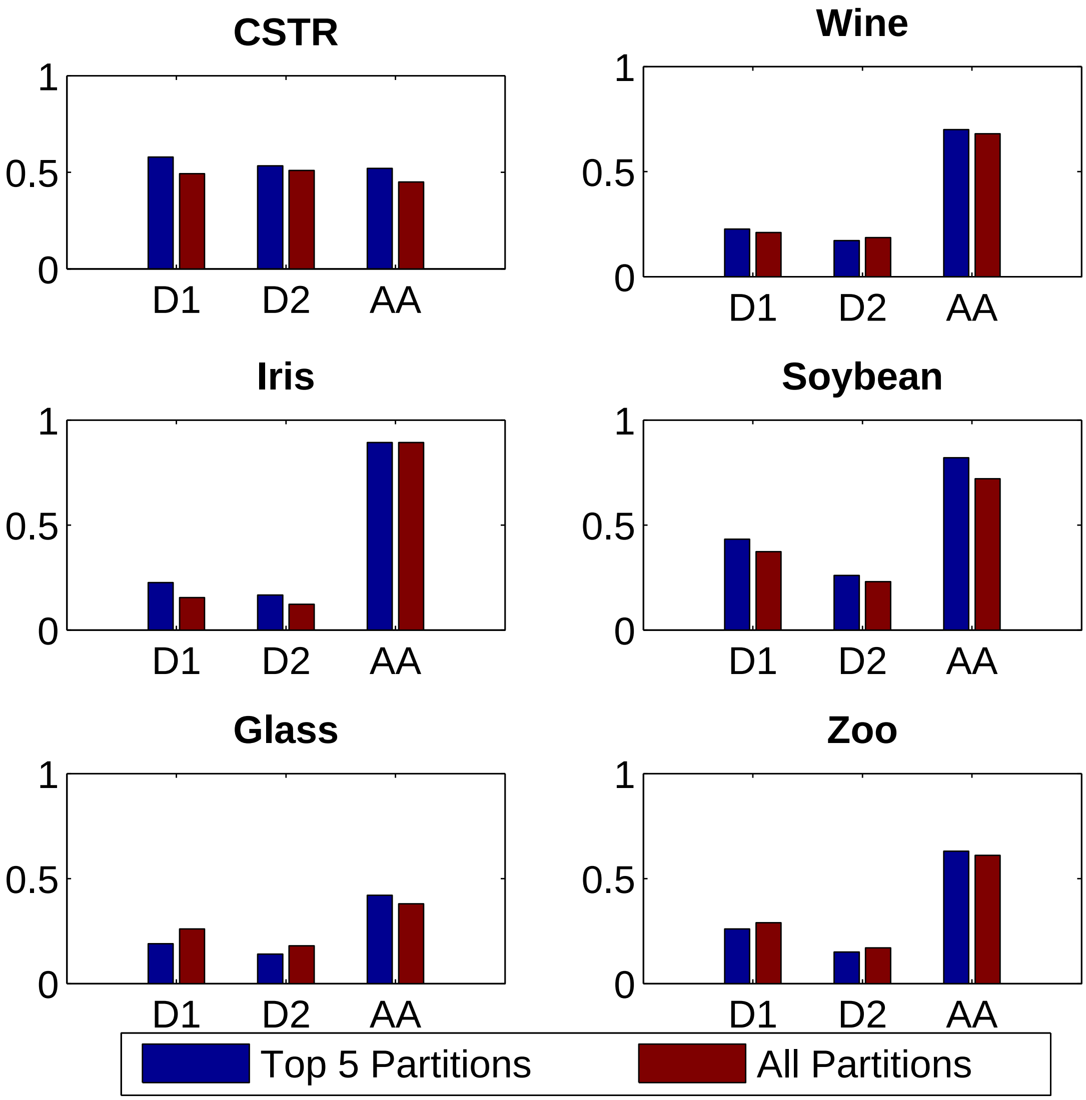


Figure 4: Diversity comparisons on the data sets. *D*1 and *D*2 represent the two diversity measures and *AA* represents the average clustering accuracy of the corresponding clusterings.

6.3.1 Heterogeneous Feature Sets

The heterogeneous feature sets include text-based feature sets and content-based feature sets. The text-based feature set consists of four components: bag-of-words features, part-of-speech statistics, lexical features and orthographic features. The content-based feature set consists of timbral features such as mel-frequency cepstral coefficients, short-term Fourier transform features, and wavelet coefficient histograms. A more detailed description of these feature sets can be found in [18].

6.3.2 Dataset Description

Our experiments are performed on a dataset consisting of 570 songs from 53 albums of a total of 41 artists. For all songs the sound recordings and the lyrics are available. Both text-based feature and content-based features are extracted for each song. To obtain the ground truth of song styles, we choose to use the similarity information between artists available at All Music Guide artist pages (http://www.allmusic.com), assuming that this information is the reflection of multiple individual users.

6.3.3 Analysis of Results

We compare the results of weighted consensus clustering with the results of NMF-based consensus clustering and the results obtained when the K-means clustering is applied separately on the two sources of data. We also compare with the following clustering strategies for integrating different information sources: (1) Feature-Level Integration: We perform K-means clustering after simply concatenating the features obtained from the two data sources. (2) Sequential Integration: We first perform clustering on one data source and obtain a clustering assignment, say, $C^1,...,C^{k_1}$. We then represent each point d_i as a k_1-dimensional vector $d_i = (d_{i1}, \cdots, d_{ik_1})$, where $d_{ij} = \begin{cases} 1 & d_i \in C^{k_j} \\ 0 & \text{otherwise} \end{cases}$, and combine the new representation with another data source using feature integration. Clustering can thus be performed on the new concatenated vectors. Depending on the order of the two sources, we have two sequential integration strategies: (2a) Sequential Integration I: first cluster based on content, then integrate with lyrics; and (2b) Sequential Integration II: first cluster based on lyrics, then integrate with content.

Table 4 presents the experimental results. From the table, we observe the following:

- The accuracy of feature-level integration is worse than that of content-only and lyric-only clustering methods. This shows that even though the joint feature space is in principle more informative than that available from individual sources, naive feature integration tends to generalize poorly [24].

- The results of sequential integration are generally better than feature-level integration, and they are comparable with those of content-only and lyrics-only.

- The weighted consensus clustering algorithms outperform all other methods. The weight clustering achieves the highest accuracy 0.471 while the NMF-based consensus clustering achieves the second highest accuracy 0.447.

Feature Set(s)	Accuracy
Content-only	0.438
Lyrics-only	0.402
Feature-Level Integration	0.380
Sequential Integration I	0.434
Sequential Integration II	0.407
NMF-based Clustering Aggregation	0.447
Weight Consensus Clustering	**0.471**

Table 4: Performance comparisons on the music dataset. The numbers are obtained by averaging over ten trials.

7 Conclusions

In this paper, we proposed a new framework for consensus clustering: the weighted consensus clustering. In this new framework, a weight for each input clustering was introduced, but the weights were automatically determined. We also showed that the weight optimization part is equivalent to the L_1 norm regression problem of LASSO and only a subset of the input clusterings contributes to the final consensus clustering. Furthermore, we demonstrated that the weights obtained could be useful for selecting input clusterings. Experiments were conducted to evaluate the effectiveness of our framework.

8 Acknowledgments

Tao Li is partially supported by IBM Faculty Research Awards, NSF CAREER Award IIS-0546280, and NIH/NIGMS S06 GM008205. Chris Ding is supported in part by a University of Texas STARS Award. We would also like to thank Yi Zhang for helping us with the experiments on measuring consensus diversity.

References

[1] A.L.N.Fred and A.K. Jain. Robust data clustering. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[2] A. D'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 2006.

[3] Marcilio C. P. de Souto, Daniel S. A. de Araujo, and Bruno L.C. da Silva. Cluster ensemble for gene expression microarray data: Accuracy and diversity. In *IJCNN*, 2006.

[4] C. Ding and X. He. K-means clustering and principal component analysis. *ICML*, 2004.

[5] C. Ding, T. Li, and M.I. Jordan. Convex and semi-nonnegative matrix factorization. *LBNL Tech Report 60428*, 2006.

[6] Chris Ding, Xiaofeng He, Hongyuan Zha, and Horst D. Simon. Adaptive dimension reduction for clustering high dimensional data. In *ICDM*, 2002.

[7] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[8] D.P.W.Ellis, B. Whitman, A. Berenzweig, and S. Lawrence. The quest for ground truth in musical artist similarity. In *Proceedings of 3rd International Conference on Music Information Retrieval*, pages 170–177, 2002.

[9] Xiaoli Zhang Fern and Carla E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *ICML*, 2004

[10] Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. Clustering aggregation. In *ICDE*, pages 341–352, 2005.

[11] Stefan T. Hadjitodorov, Ludmila I. Kuncheva, and Ludmila P. Todorova. Moderate diversity for better cluster ensembles. *Information Fusion*, 7(3):264–275, 2006.

[12] E-H. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. WebACE: A web agent for document categorization and exploration. In *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*. ACM Press, 1998.

[13] Xiaohua Hu, Illhoi Yoo, Xiaodan Zhang, Payal Nanavati, and Debjit Das. Wavelet transformation and cluster ensemble for gene expression analysis. *International Journal of Bioinformatics Research and Application*, 1(4):447–460, 2006.

[14] G. R. G. Lanckriet, N. Cristianini, P.L. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semi-definite programming. *ICML*, pages 323–330, 2006.

[15] D.D. Lee and H.S. Seung. Learning the parts of objects with nonnegative matrix factorization. *Nature*, 401:788–791, 1999.

[16] Tao Li, Chris Ding, and Michael I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *ICDM*, 2007.

[17] Tao Li, Mitsunori Ogihara, and Sheng Ma. On combining multiple clusterings. In *CIKM*, pages 294–303, 2004.

[18] Tao Li, Mitsunori Ogihara, and Shenghuo Zhu. Integrating features from different sources for music information retrieval. In *ICDM*, pages 372–381, 2006.

[19] Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow, 1996.

[20] G.W. Milligan and M.C. Cooper. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivar Behav Res*, 21:846–850, 1986.

[21] Lance Parsons, Ehtesham Haque, and Huan Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explor. Newsl.*, 6(1):90–105, 2004.

[22] Alexander Strehl and Joydeep Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research (JMLR)*, 3:583–617, December 2002.

[23] R. Tibshirani. Regression shrinkage and selection via the LASSO. *J. Royal. Statist. Soc B.*, 58:267–288, 1996.

[24] Lizhong Wu, Sharon L. Oviatt, and Philip R. Cohen. Multimodal integration - a statistical view. *IEEE Transactions on Multimedia*, 1(4):334–341, 1999.

[25] H. Zha, C. Ding, M. Gu, X. He, and H.D. Simon. Spectral relaxation for K-means clustering. *Advances in Neural Information Processing Systems 14 (NIPS'01)*, pages 1057–1064, 2002.

[26] Z. Zhang, H. Zha, and H.D. Simon. Low-rank approximations with sparse factors ii: Penalized methods with discrete newton-like iterations. *SIAM J. Matrix Analysis Applications*, 25:901–920, 2004.

[27] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *J. Computational and Graphical Statistics*, 15:265–286, 2006.