# A Theoretic Framework of K-Means-Based Consensus Clustering

**Junjie Wu**[1], **Hongfu Liu**[1], **Hui Xiong**[2], **Jie Cao**[3]
[1]School of Economics and Management, Beihang University
[2]Rutgers Business School, Rutgers University
[3]Jiangsu Provincial Key Lab. of E-Business, Nanjing Univ. of Fin. and Eco.
wujj@buaa.edu.cn, lauhf@sem.buaa.edu.cn, hxiong@rutgers.edu, jie.cao@njue.edu.cn

## Abstract

Consensus clustering emerges as a promising solution to find cluster structures from data. As an efficient approach for consensus clustering, the K-means based method has garnered attention in the literature, but the existing research is still preliminary and fragmented. In this paper, we provide a systematic study on the framework of K-means-based Consensus Clustering (KCC). We first formulate the general definition of KCC, and then reveal a necessary and sufficient condition for utility functions that work for KCC, on both complete and incomplete basic partitionings. Experimental results on various real-world data sets demonstrate that KCC is highly efficient and is comparable to the state-of-the-art methods in terms of clustering quality. In addition, KCC shows high robustness to incomplete basic partitionings with substantial missing values.

## 1 Introduction

*Consensus clustering*, also known as *cluster ensemble*, aims to find a single partitioning of data from multiple existing *basic partitionings* [Monti *et al.*, 2003; Zhang *et al.*, 2010]. It has been recognized that consensus clustering may help to generate robust partitionings, find bizarre clusters, handle noise and outliers, and integrate solutions from multiple distributed sources [Nguyen and Caruana, 2007].

Consensus clustering is NP-complete in essence [Filkov and Steven, 2004a; Topchy *et al.*, 2005]. In the literature, many algorithms have been proposed to address the computational challenges, among which a K-means-based method proposed in [Topchy *et al.*, 2003] attracts great interests. However, the general theory for utility functions that work for *K-means-based Consensus Clustering* (KCC) is yet unavailable, which prevents KCC from being widely used.

In this paper, we provided a systematic study on the theoretic framework of K-means-based consensus clustering. The major contributions are summarized as follows. First, we formally defined the concept of KCC, and provided a necessary and sufficient condition for the so-called *KCC utility functions*, which establishes the general KCC framework. Second, we redesigned the computational procedures for KCC utility functions and K-means clustering, and thus extended the framework to the more general scenario where incomplete basic partitionings present. Third, we proposed the standard and normalized forms of a KCC utility function, and gave some particular examples for practical uses.

Extensive experiments on various real-world data sets demonstrated that KCC is highly efficient and is comparable to the state-of-the-art methods in terms of clustering quality. We found that: a) While multiple utility functions can improve the usability of KCC on different types of data, the ones based on Shannon entropy generally exhibit more robust performances; b) The generation strategy of basic partitionings is a critical factor that influences the performance of KCC; c) KCC is very robust with even very few basic partitionings of high quality, or on severely incomplete basic partitionings.

## 2 Related Work

Consensus clustering (CC) is essentially a combinatorial optimization problem. The existing literature can be roughly divided into two categories: CC with implicit objectives (CCIO) and CC with explicit objectives (CCEO).

Methods in CCIO do not set global objective functions. Rather, they directly adopt some heuristics to find approximate solutions. The representative methods include the graph-based algorithms [Strehl and Ghosh, 2002; Fern and Brodley, 2004], the co-association matrix based methods [Fred and Jain, 2005; Wang *et al.*, 2009], Relabeling and Voting methods [Fischer and Buhmann, 2003; Ayad and Kamel, 2008], Locally Adaptive Cluster based methods [Domeniconi and Al-Razgan, 2009], genetic algorithm based methods [Yoon *et al.*, 2006], and still many more.

Methods in CCEO have explicit global objective functions for consensus clustering. The Median Partition problem based on Mirkin distance is among the oldest ones [Filkov and Steven, 2004b; Gionis *et al.*, 2007]. In the inspiring work, [Topchy *et al.*, 2003] proposed a Quadratic Mutual Information based objective function and used K-means clustering to find the solution. This elegant idea could be traced back to the work by Mirkin on the Category Utility Function [Mirkin, 2001]. Other solutions for different objective functions include EM algorithm [Topchy *et al.*, 2004], nonnegative matrix factorization [Li *et al.*, 2007], kernel-based methods [Vega-Pons *et al.*, 2010], simulated annealing [Lu *et al.*, 2008], and among others.

More algorithms for consensus clustering could be found in some survey papers [Ghaemi *et al.*, 2009; Vega-Pons and Ruiz-shulcloper, 2011; Li *et al.*, 2010]. In general, compared with CCIO methods, CCEO methods might offer better interpretability and higher robustness to clustering results, via the guidance of objective functions. However, they often bear high computational costs. Moreover, one CCEO method typically works for one objective function, which seriously limits its applicative scope. These indeed motivate our study in this paper. We attempt to build a general theoretic framework for efficient K-means-based consensus clustering using multiple utility functions.

## 3 Problem Formulation

We begin by introducing some basic mathematical notations. Let $\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$ denote a set of data objects. A partitioning of $\mathcal{X}$ into $K$ crisp clusters can be represented as a collection of $K$ subsets of objects in $\mathcal{C} = \{C_k | k = 1, \cdots, K\}$, with $C_k \bigcap C_{k'} = \emptyset$, $\forall k \neq k'$, and $\bigcup_{k=1}^{K} C_k = \mathcal{X}$, or as a label vector $\pi = (L_\pi(x_1), \cdots, L_\pi(x_n))'$, where $L_\pi(x_i)$ maps $x_i$ to one of the $K$ labels: $\{1, \cdots, K\}$.

Given $r$ existed partitionings, i.e., *basic partitionings*, of $\mathcal{X}$ in $\Pi = \{\pi_1, \pi_2, \cdots, \pi_r\}$, the goal of consensus clustering is formulated as to find a consensus partitioning $\pi$ such that

$$\Gamma(\pi, \Pi) = \sum_{i=1}^{r} w_i U(\pi, \pi_i) \tag{1}$$

is maximized, where $\Gamma : \mathbb{Z}_{++}^n \times \mathbb{Z}_{++}^{nr} \mapsto \mathbb{R}$ is a *consensus function*, $U : \mathbb{Z}_{++}^n \times \mathbb{Z}_{++}^n \mapsto \mathbb{R}$ is a *utility function*, and $w_i \in [0, 1]$ is a user-specified *weight* for $\pi_i$, with $\sum_{i=1}^{r} w_i = 1$. Apparently, the choice of the utility function in Eq. (1) is crucial to the success of consensus clustering, and largely determines the heuristics to employ.

### 3.1 K-means-based Consensus Clustering

Though being more recognized in the clustering field [Jain and Dubes, 1988], K-means [MacQueen, 1967] is essentially a very fast and robust heuristic for combinatorial optimization. A natural idea is, can we use K-means to optimize Eq. (1)? Or equivalently, can we transform the consensus clustering problem equivalently to a much simpler K-means clustering problem?

We first restrict the computation of $U$ to the *contingency matrix*, a co-occurrence matrix for the clusters of two partitionings. Suppose we have $\pi$ and $\pi_i$, the consensus partitioning and the $i$th basic partitioning with $K$ and $K_i$ clusters, respectively. The $i$th contingency matrix is defined as $M^{(i)} = [n_{kj}^{(i)}]_{K \times K_i}$, where $n_{kj}^{(i)}$ denotes the number of data objects in both cluster $C_k$ of $\pi$ and cluster $C_j^{(i)}$ of $\pi_i$, $\forall\, k, j$. The summations can thus be given by having $n_{k+} = \sum_{j=1}^{K_i} n_{kj}^{(i)}$ and $n_{+j}^{(i)} = \sum_{k=1}^{K} n_{kj}^{(i)}$, $\forall\, k, j$. Let $p_{kj}^{(i)} = n_{kj}^{(i)}/n$, $p_{k+} = n_{k+}/n$, and $p_{+j}^{(i)} = n_{+j}^{(i)}/n$, we then have the *normalized contingency matrix* $M_n^{(i)}$, $\forall\, i$, based on which a wide range of utility functions can be defined accordingly.

K-means does not work on the contingency matrix. As inspired by [Topchy *et al.*, 2003], a *binary matrix* is introduced for it. Let $\mathcal{X}^{(b)} = \{x_l^{(b)} | 1 \leq l \leq n\}$ be a binary data set derived from $r$ basic partitionings in $\Pi$ such that $x_l^{(b)} = (x_{l,1}^{(b)}, \cdots, x_{l,r}^{(b)})$, $x_{l,i}^{(b)} = (x_{l,i1}^{(b)}, \cdots, x_{l,iK_i}^{(b)})$, and

$$x_{l,ij}^{(b)} = \begin{cases} 1, & \text{if } L_{\pi_i}(x_l) = j \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

Accordingly, $\mathcal{X}^{(b)}$ is an $n \times \sum_{i=1}^{r} K_i$ binary matrix with $|x_{l,i}^{(b)}| = 1$, $\forall\, l, i$. We can thus employ K-means on $\mathcal{X}^{(b)}$ to generate one partitioning $\pi'$ with $K$ clusters. If $\pi' = \pi$ in Eq. (1), it means we can transform consensus clustering to K-means clustering equivalently, and thus enable the *K-means-based Consensus Clustering* (KCC) scheme.

Let $m_k = (m_{k,1}, \cdots, m_{k,r})$ denote the centroid of the $k$th cluster of $\pi'$, with $m_{k,i} = (m_{k,i1}, \cdots, m_{k,iK_i})$. We then have the following definition:

**Definition 1** *A utility function $U$ is a KCC Utility Function, if $\forall\, \Pi = \{\pi_1, \cdots, \pi_r\}$ and $K \geq 2$, there exists a distance function $f$ such that*

$$\max_{\pi \in F} \sum_{i=1}^{r} w_i U(\pi, \pi_i) \Longleftrightarrow \min_{\pi \in F} \sum_{k=1}^{K} \sum_{x_l^{(b)} \in C_k} f(x_l^{(b)}, m_k) \tag{3}$$

*holds for any feasible region $F$.*

Accordingly, the critical point of KCC is to identify KCC utility functions ($U$) and their corresponding distance functions ($f$) so that the left-hand-side problem in Eq. (3) can turn into the right-hand-side problem.

## 4 General Theory of KCC Utility Function

In this section, we propose the necessary and sufficient condition for being a KCC utility function, and thus establish the general framework of KCC. We also extend the theory to the situation where incomplete basic partitionings exist.

### 4.1 The Necessary and Sufficient Condition

Here we formulate how a utility function can be a KCC utility function. A lemma is first given as follows:

**Lemma 1** *A utility function $U$ is a KCC utility function, if and only if $\forall\, \Pi$ and $K \geq 2$, there exist a differentiable convex function $\phi$ and a strictly increasing function $g_{\Pi,K}$ such that*

$$\sum_{i=1}^{r} w_i U(\pi, \pi_i) = g_{\Pi,K}\Big(\sum_{k=1}^{K} p_{k+} \phi(m_k)\Big). \tag{4}$$

PROOF. We first give one fact as follows. Suppose $f$ is a distance function that fits K-means clustering. Then according to [Wu *et al.*, 2012], $f : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ is a point-to-centroid distance that can be derived by a continuously differentiable convex function $\phi$:

$$f(x, y) = \phi(x) - \phi(y) - (x - y)^T \nabla \phi(y). \tag{5}$$

Accordingly, we have

$$\sum_{k=1}^{K}\sum_{x_l^{(b)}\in C_k} f(x_l^{(b)}, m_k) = \sum_{x_l^{(b)}\in \mathcal{X}^{(b)}} \phi(x_l^{(b)}) - n\sum_{k=1}^{K} p_{k+}\phi(m_k). \tag{6}$$

Since both $\sum_{x_l^{(b)}\in \mathcal{X}^{(b)}} \phi(x_l^{(b)})$ and $n$ are constants given $\Pi$,

$$\min_{\pi}\sum_{k=1}^{K}\sum_{x_l^{(b)}\in C_k} f(x_l^{(b)}, m_k) \Longleftrightarrow \max_{\pi}\sum_{k=1}^{K} p_{k+}\phi(m_k). \tag{7}$$

Now let us turn back to the proof of the sufficient condition. As $g_{\Pi,K}$ is strictly increasing, we have

$$\max_{\pi}\sum_{i=1}^{r} w_i U(\pi,\pi_i) \Longleftrightarrow \max_{\pi}\sum_{k=1}^{K} p_{k+}\phi(m_k). \tag{8}$$

So we finally have Eq. (3), which indicates that $U$ is a KCC utility function. The sufficient condition holds.

We then prove the necessary condition. Suppose the distance function $f$ in Eq. (3) is derived from a differentiable convex function $\phi$. By Eq. (3) and Eq. (7), we have Eq. (8).

Let $\Upsilon(\pi)$ denote $\sum_{i=1}^{r} w_i U(\pi,\pi_i)$ and $\Psi(\pi)$ denote $\sum_{k=1}^{K} p_{k+}\phi(m_k)$ for convenience. Note that Eq. (8) holds for any feasible region $F$ because Eq. (7) is derived from the equality rather than equivalence relationship in Eq. (6). Therefore, for any two consensus partitionings $\pi'$ and $\pi''$, if we let $F = \{\pi', \pi''\}$, we have

$$\Upsilon(\pi') > (=, \text{or} <)\ \Upsilon(\pi'') \iff \Psi(\pi') > (=, \text{or} <)\ \Psi(\pi''), \tag{9}$$

which indicates that $\Upsilon$ and $\Psi$ have the same ranking over all the possible partitionings in the universal set $\mathcal{F} = \{\pi | L_\pi(x_l^{(b)}) \in \{1,\cdots,K\}, 1 \le l \le n\}$. Define a mapping $g_{\Pi,K}$ that maps $\Psi(\pi)$ to $\Upsilon(\pi)$, $\pi \in \mathcal{F}$. According to Eq. (9), $g_{\Pi,K}$ is a function, and $\forall\, x > x',\ g_{\Pi,K}(x) > g_{\Pi,K}(x')$. This implies that $g_{\Pi,K}$ is strictly increasing. So the necessary condition holds, and the lemma thus follows. ■

We here use $\Pi$ and $K$ as the subscripts for $g$, because these parameters directly affect the ranking of $\pi$ in $\mathcal{F}$ given by $\Upsilon$ or $\Psi$. That is, different mapping functions may exist for different settings of $\Pi$ and $K$. Next, we go a further step to analyze Eq. (4). Recall the contingency matrix mentioned in Section 3.1. Let $P_k^{(i)} = (p_{k1}^{(i)}/p_{k+}, p_{k2}^{(i)}/p_{k+}, \cdots, p_{kK_i}^{(i)}/p_{k+}), \forall\, i,k$. We give without proof the following lemma:

**Lemma 2** *For K-means clustering on $\mathcal{X}^{(b)}$, the centroids*

$$m_{k,i} = \left(\frac{p_{k1}^{(i)}}{p_{k+}}, \cdots, \frac{p_{kj}^{(i)}}{p_{k+}}, \cdots, \frac{p_{kK_i}^{(i)}}{p_{k+}}\right) = P_k^{(i)}, \forall\, k, i. \tag{10}$$

We then have the following important theorem:

**Theorem 1** *$U$ is a KCC utility function, if and only if $\forall\ \Pi = \{\pi_1,\cdots,\pi_r\}$ and $K \ge 2$, there exists a set of continuously differentiable convex functions $\{\mu_1,\cdots,\mu_r\}$ such that*

$$U(\pi,\pi_i) = \sum_{k=1}^{K} p_{k+}\mu_i(P_k^{(i)}), \forall\, i. \tag{11}$$

*The convex function $\phi$ for the corresponding K-means clustering is given by*

$$\phi(m_k) = \sum_{i=1}^{r} w_i \nu_i(m_{k,i}), \forall\, k, \tag{12}$$

*where*

$$\nu_i(x) = a\mu_i(x) + c_i, a \in \mathbb{R}_{++}, c_i \in \mathbb{R}, \forall\, i. \tag{13}$$

PROOF. We first prove the sufficient condition. If we substitute $U(\pi,\pi_i)$ in Eq. (11) into the left-hand-side of Eq. (4) and use Eq. (12) and Eq. (13), we have

$$\sum_{i=1}^{r} w_i U(\pi,\pi_i) = \frac{1}{a}\sum_{k=1}^{K} p_{k+}\phi(m_k) - \frac{1}{a}\sum_{i=1}^{r} w_i c_i. \tag{14}$$

Let $g_{\Pi,K}(x) = a_\Pi x + b_\Pi$, with $a_\Pi = \frac{1}{a}$ and $b_\Pi = -\frac{1}{a}\sum_{i=1}^{r} w_i c_i$. $g_{\Pi,K}$ is thus a strictly increasing function for $a > 0$. We then have $\sum_{i=1}^{r} w_i U(\pi,\pi_i) = g_{\Pi,K}(\sum_{k=1}^{K} p_{k+}\phi(m_k))$. $U$ is thus a KCC utility function by Lemma 1. The sufficient condition follows.

We then prove the necessary condition. Due to the arbitrariness of $\Pi$ in Eq. (4), we can let $\Pi \doteq \Pi_i = \{\pi_i\}, \forall\, i$. Accordingly, $m_k$ reduces to $m_{k,i}$, and $\phi(m_{k,i})$ reduces to $\phi_i(m_{k,i})$, i.e., the $\phi$ function defined only on the $i$th "block" of $m_k$ without involvement of the weight $w_i$. Then by Eq. (4), we have

$$U(\pi,\pi_i) = g_{\Pi_i,K}(\sum_{k=1}^{K} p_{k+}\phi_i(m_{k,i})),\ 1 \le i \le r, \tag{15}$$

where $g_{\Pi_i,K}$ is the mapping function when $\Pi$ reduces to $\Pi_i$. By summing up $U(\pi,\pi_i)$ from $i=1$ to $r$, we have

$$\sum_{i=1}^{r} w_i U(\pi,\pi_i) = \sum_{i=1}^{r} w_i\ g_{\Pi_i,K}(\sum_{k=1}^{K} p_{k+}\phi_i(m_{k,i})),$$

which, in comparison to Eq. (4), indicates that

$$g_{\Pi,K}(\sum_{k=1}^{K} p_{k+}\phi(m_k)) = \sum_{i=1}^{r} w_i g_{\Pi_i,K}(\sum_{k=1}^{K} p_{k+}\phi_i(m_{k,i})). \tag{16}$$

If we take the partial derivative with respect to $m_{k,ij}$ on both sides, we have

$$\underbrace{g'_{\Pi,K}(\sum_{k=1}^{K} p_{k+}\phi(m_k))}_{(\alpha)}\underbrace{\frac{\partial\phi(m_k)}{\partial m_{k,ij}}}_{(\beta)} = w_i\ \underbrace{g'_{\Pi_i,K}(\sum_{k=1}^{K} p_{k+}\phi_i(m_{k,i}))}_{(\gamma)}\underbrace{\frac{\partial\phi_i(m_{k,i})}{\partial m_{k,ij}}}_{(\delta)}. \tag{17}$$

As $(\gamma)$ and $(\delta)$ do not contain any weight parameters $w_l$, $1 \le l \le r$, the right-hand-side of Eq. (17) has one and only one weight parameter: $w_i$. This implies that $(\alpha)$ is a constant, otherwise the left-hand-side of Eq. (17) would contain multiple weight parameters other than $w_i$ because of the existence of $\phi(m_k)$ in $g'_{\Pi,K}$. Analogously, since $(\beta)$ does not contain all $p_{k+}$, $1 \le k \le K$, $(\gamma)$ must also be a constant. These

results imply that $g_{\Pi,K}$ and $g_{\Pi_i,K}$, $1 \leq i \leq r$, are all linear functions. Without loss of generality, we let

$$g_{\Pi,K}(x) = a_\Pi x + b_\Pi,\ \forall\, a_\Pi \in \mathbb{R}_{++}, b_\Pi \in \mathbb{R},\ \text{and} \quad (18)$$

$$g_{\Pi_i,K}(x) = a_i x + b_i,\ \forall\, a_i \in \mathbb{R}_{++}, b_i \in \mathbb{R}. \quad (19)$$

Eq. (17) thus turns into

$$a_\Pi \frac{\partial \phi(m_k)}{\partial m_{k,ij}} = w_i a_i \frac{\partial \phi_i(m_{k,i})}{\partial m_{k,ij}},\ \forall\, i, j. \quad (20)$$

$\partial \phi_i(m_{k,i})/\partial m_{k,ij}$ is the function of $\{m_{k,i1}, \cdots, m_{k,iK_i}\}$ only, which implies that $\partial \phi(m_k)/\partial m_{k,ij}$ is not the function of $m_{k,l}, \forall\, l \neq i$. As a result, given the arbitrariness of $i$, we have $\phi(m_k) = \varphi(\phi_1(m_{k,1}), \cdots, \phi_r(m_{k,r}))$, which indicates

$$\frac{\partial \phi(m_k)}{\partial m_{k,ij}} = \frac{\partial \varphi}{\partial \phi_i} \frac{\partial \phi_i(m_{k,i})}{\partial m_{k,ij}}.$$

Accordingly, Eq. (20) turns into $\partial \varphi/\partial \phi_i = w_i a_i / a_\Pi$, which leads to

$$\phi(m_k) = \sum_{i=1}^{r} \frac{w_i a_i}{a_\Pi} \phi_i(m_{k,i}) + d_i, \forall\, d_i \in \mathbb{R}. \quad (21)$$

Let $\nu_i(x) = \frac{a_i}{a_\Pi}\phi_i(x) + \frac{d_i}{w_i}$, $1 \leq i \leq r$, Eq. (12) thus follows.

Moreover, according to Eq. (15) and Eq. (19), we have

$$U(\pi, \pi_i) = \sum_{k=1}^{K} p_{k+}(a_i \phi_i(P_k^{(i)}) + b_i), \forall\, i. \quad (22)$$

Let $\mu_i(x) = a_i \phi_i(x) + b_i$, $1 \leq i \leq r$, Eq. (11) follows. Further let $a = 1/a_\Pi$ and $c_i = d_i/w_i - b_i/a_\Pi$, we also have Eq. (13). The necessary condition thus follows. ■

Theorem 1 gives the general necessary and sufficient condition for being a KCC utility function. That is, a KCC utility function must be a weighted average of a set of convex functions defined on $P_k^{(i)}$, $1 \leq i \leq r$, respectively. From this perspective, Theorem 1 can serve as the criterion to verify whether a given utility function is a KCC utility function or not. Nevertheless, the most important thing is, Theorem 1 indicates the way to conduct the K-means-based consensus clustering. That is, we first design or designate a set of convex functions $\mu_i$ defined on $P_k^{(i)}$, $1 \leq i \leq r$, from which the utility function as well as the consensus function can be derived by Eq. (11) and Eq. (1), respectively; then after setting $a$ and $c_i$ in Eq. (13), we can determine the corresponding $\phi$ for K-means clustering by Eq. (12), which is further used to derive the point-to-centroid distance $f$ using Eq. (5); the K-means clustering is finally employed to find the consensus partitioning $\pi$. In practice, we often simplify the above procedure by letting $a = 1, c_i = 0, \mu \equiv \mu_i \equiv \nu_i, \forall\, i$, which also become the default settings in the experiments to follow.

## 4.2 Two Forms of KCC Utility Functions

Let $\mu \equiv \mu_i, \forall\, i$, in Eq. (11). We denote the KCC utility function derived by $\mu$ in Eq. (11) as $U_\mu$ for short. In what follows, we introduce two special forms of KCC utility functions given any $U_\mu$ (or equivalently $\mu$).

*Standard Form.* Let $P^{(i)} = (p_{+1}^{(i)}, \cdots, p_{+K_i}^{(i)})$, and let $\mu_s(P_k^{(i)}) = \mu(P_k^{(i)}) - \mu(P^{(i)})$. Then by Eq. (11), we obtain a new utility function $U_{\mu_s}$ as follows:

$$U_{\mu_s}(\pi, \pi_i) = U_\mu(\pi, \pi_i) - \mu(P^{(i)}). \quad (23)$$

As $\mu(P^{(i)})$ is a constant given $\pi_i$, $\mu_s$ and $\mu$ will lead to a same point-to-centroid distance $f$, and thus a same consensus partitioning $\pi$. It is easy to show $U_{\mu_s}$ is non-negative, and thus can be viewed as the *utility gain* from a consensus clustering. We here define $U_{\mu_s}$ as the *standard form* of $U_\mu$. If we further let $\mu_{ss}(P_k^{(i)}) = \mu_s(P_k^{(i)}) - \mu_s(P^{(i)})$, we have $\mu_{ss} = \mu_s \Rightarrow U_{\mu_{ss}} = U_{\mu_s}$. This implies that any KCC utility function has one and only one standard form.

*Normalized Form.* Let $\mu_n(P_k^{(i)}) = \mu_s(P_k^{(i)})/|\mu(P^{(i)})|$, we have a new KCC utility function as follows:

$$U_{\mu_n}(\pi, \pi_i) = \frac{U_{\mu_s}(\pi, \pi_i)}{|\mu(P^{(i)})|} = \frac{U_\mu(\pi, \pi_i) - \mu(P^{(i)})}{|\mu(P^{(i)})|}. \quad (24)$$

Apparently, $U_{\mu_n} \geq 0$, which can be viewed as the *utility gain ratio* to the constant $|\mu(P^{(i)})|$, and thus is called the *normalized form* of $U_\mu$. It is noteworthy that a KCC utility function also has one and only one normalized form.

In summary, given a convex function $\mu$, we can derive a KCC utility function $U_\mu$, as well as its only standard form $U_{\mu_s}$ and normalized form $U_{\mu_n}$. Given clear physical meanings, the standard form and normalized form will be adopted as two major forms of KCC utility functions in the experimental section below. Table 1 shows some sample KCC utility functions, which are all in their standard forms.

## 4.3 Handling Incomplete Basic Partitionings

Here, we introduce how to conduct K-means-based consensus clustering with the presence of incomplete basic partitionings (IBPs). An IBP $\pi_i$ is obtained by clustering a data *subset* $\mathcal{X}_i \subseteq \mathcal{X}$, $1 \leq i \leq r$, with the constraint that $\bigcup_{i=1}^{r} \mathcal{X}_i = \mathcal{X}$. The problem is, given $r$ IBPs in $\Pi = \{\pi_1, \cdots, \pi_r\}$, how to cluster $\mathcal{X}$ into $K$ crisp clusters using KCC?

We first adjust the way for utility computation on IBPs. We still have maximizing Eq. (1) as the objective function, but modify the normalized contingency matrix $M_n^{(i)}$ so that: $n^{(i)} = |\mathcal{X}_i|$, $p_{kj}^{(i)} = n_{kj}^{(i)}/n^{(i)}$, $p_{k+}^{(i)} = n_{k+}^{(i)}/n^{(i)}$, $p_{+j}^{(i)} = n_{+j}^{(i)}/n^{(i)}$, and $p^{(i)} = n^{(i)}/n, \forall\, j, k, i$.

We then adjust K-means clustering for the incomplete binary data set $\mathcal{X}^{(b)}$. Let the distance $f$ be the sum of the distances on different basic partitionings, i.e.,

$$f(x_l^{(b)}, m_k) = \sum_{i=1}^{r} I(x_l \in \mathcal{X}_i) f_i(x_{l,i}^{(b)}, m_{k,i}), \quad (25)$$

where $f_i$ is $f$ on the $i$th "block" of $\mathcal{X}^{(b)}$, and $I(x_l \in \mathcal{X}_i) = 1$ if $x_l \in \mathcal{X}_i$ and 0 otherwise. Further let the centroid $m_{k,i} = (m_{k,i1}, \cdots, m_{k,iK_i})$, with

$$m_{k,ij} = \frac{\sum_{x_l \in C_k \bigcap \mathcal{X}_i} x_{l,ij}^{(b)}}{|C_k \bigcap \mathcal{X}_i|} = \frac{n_{kj}^{(i)}}{n_{k+}^{(i)}} = \frac{p_{kj}^{(i)}}{p_{k+}^{(i)}},\ \forall\, k, i, j. \quad (26)$$

In what follows, we give without proof a theorem about the convergence of K-means clustering on incomplete $\mathcal{X}^{(b)}$:

Table 1: Sample KCC Utility Functions

| Notation | $\mu(m_{k,i})$ | $U_\mu(\pi, \pi_i)$ | $f(x_l^{(b)}, m_k)$ |
|---|---|---|---|
| $U_c$ | $\|m_{k,i}\|_2^2 - \|P^{(i)}\|_2^2$ | $\sum_{k=1}^K p_{k+}\|P_k^{(i)}\|_2^2 - \|P^{(i)}\|_2^2$ | $\sum_{i=1}^r w_i \|x_{l,i}^{(b)} - m_{k,i}\|_2^2$ |
| $U_H$ | $(-H(m_{k,i})) - (-H(P^{(i)}))$ | $\sum_{k=1}^K p_{k+}(-H(P_k^{(i)})) - (-H(P^{(i)}))$ | $\sum_{i=1}^r w_i D(x_{l,i}^{(b)} \| m_{k,i})$ |
| $U_{\cos}$ | $\|m_{k,i}\|_2 - \|P^{(i)}\|_2$ | $\sum_{k=1}^K p_{k+}\|P_k^{(i)}\|_2 - \|P^{(i)}\|_2$ | $\sum_{i=1}^r w_i(1 - \cos(x_{l,i}^{(b)}, m_{k,i}))$ |
| $U_{L_p}$ | $\|m_{k,i}\|_p - \|P^{(i)}\|_p$ | $\sum_{k=1}^K p_{k+}\|P_k^{(i)}\|_p - \|P^{(i)}\|_p$ | $\sum_{i=1}^r w_i(1 - \frac{\sum_{j=1}^{K_i} x_{l,ij}^{(b)}(m_{k,ij})^{p-1}}{\|m_{k,i}\|_p^{p-1}})$ |

Note: $\|x\|_p$ – $L_p$ norm of $x$; $H$ – Shannon entropy; $D$ – KL-divergence; cos – cosine similarity.

Table 2: Comparison of Execution Time (in seconds)

| | $brea.$ | $ecol.$ | $iris$ | $pend.$ | $sati.$ | $derm.$ | $wine$ | $mm$ | $revi.$ | $la12$ | $spor.$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $KCC(U_c)$ | **1.95** | **1.40** | 0.33 | **81.19** | **32.47** | **1.26** | 0.56 | **2.78** | **4.44** | **8.15** | **11.33** |
| $GP$ | 8.80 | 6.79 | 4.08 | N/A | 54.39 | 6.39 | 3.92 | 15.40 | 32.35 | N/A | N/A |
| $HCC$ | 18.85 | 2.33 | **0.18** | N/A | 2979.48 | 2.78 | **0.28** | 535.63 | 2154.45 | 6486.87 | N/A |

N/A: Failure due to the out-of-memory error.

Table 3: Experimental Data Sets

| Data Sets | Source | #Objects | #Attributes | #Classes |
|---|---|---|---|---|
| $breast\_w$ | UCI | 699 | 9 | 2 |
| $ecoli^\dagger$ | UCI | 332 | 7 | 6 |
| $iris$ | UCI | 150 | 4 | 3 |
| $pendigits$ | UCI | 10992 | 16 | 10 |
| $satimage$ | UCI | 4435 | 36 | 6 |
| $dermatology$ | UCI | 358 | 33 | 6 |
| $wine^\ddagger$ | UCI | 178 | 13 | 3 |
| $mm$ | TREC | 2521 | 126373 | 2 |
| $reviews$ | TREC | 4069 | 126373 | 5 |
| $la12$ | TREC | 6279 | 31472 | 6 |
| $sports$ | TREC | 8580 | 126373 | 7 |

†: two clusters containing only two objects were deleted as noise.
‡: the values of the last attribute were normalized by a scaling factor 100.

**Theorem 2** *K-means clustering using $f$ in Eq. (25) as the distance function and using $m$ in Eq. (26) as the centroid, is guaranteed to converge in finite iterations.*

Until now we can extend KCC to the IBP case. Let $P_k^{(i)} = (p_{k1}^{(i)}/p_{k+}^{(i)}, \cdots, p_{kK_i}^{(i)}/p_{k+}^{(i)}) = m_{k,i}, \forall\, k, i$. We have:

**Theorem 3** *$U$ is a KCC utility function, if and only if $\forall\ \Pi = \{\pi_1, \cdots, \pi_r\}$ and $K \geq 2$, there exists a set of continuously differentiable convex functions $\{\mu_1, \cdots, \mu_r\}$ such that*

$$U(\pi, \pi_i) = p^{(i)} \sum_{k=1}^{K} p_{k+}^{(i)} \mu_i(P_k^{(i)}), \forall\, i. \tag{27}$$

*The convex function $\phi_i$, $1 \leq i \leq r$, for the corresponding K-means clustering is given by*

$$\phi_i(m_{k,i}) = w_i \nu_i(m_{k,i}), \forall\, k, \tag{28}$$

*where*

$$\nu_i(x) = a\mu_i(x) + c_i, a \in \mathbb{R}_{++}, c_i \in \mathbb{R}, \forall\, i. \tag{29}$$

The proof is similar to the one for Theorem 1, so we omit it here. Eq. (27) is very similar to Eq. (11) except for having additional parameters $p^{(i)}, \forall\, i$. This implies that the basic partitioning on a larger data subset will have more impact on the consensus clustering, which is considered reasonable. Since Eq. (27) reduces to Eq. (11) when IBPs turn into complete BPs, Theorem 3 is indeed a more general version of Theorem 1.

# 5 Experimental Results

In this section, we present experimental results of KCC on various real-world data sets listed in Table 3.

Table 4: KCC Clustering Results (by $R_n$)

| | $U_c$ | $U_H$ | $U_{cos}$ | $U_{L5}$ | $U_{L8}$ | $NU_c$ | $NU_H$ | $NU_{cos}$ | $NU_{L5}$ | $NU_{L8}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $brea.$ | 0.06 | 0.86 | 0.11 | 0.12 | 0.13 | 0.04 | **0.87** | 0.12 | 0.13 | 0.11 |
| $ecol.$ | 0.51 | 0.43 | 0.44 | 0.44 | 0.43 | 0.50 | **0.55** | 0.42 | 0.42 | 0.43 |
| $iris$ | 0.74 | 0.73 | 0.74 | 0.74 | **0.75** | 0.73 | 0.71 | **0.75** | **0.75** | 0.74 |
| $pend.$ | 0.53 | 0.56 | **0.58** | 0.57 | 0.55 | 0.51 | 0.57 | **0.58** | 0.57 | 0.56 |
| $sati.$ | 0.45 | 0.47 | 0.53 | 0.47 | 0.48 | **0.33** | **0.53** | **0.53** | 0.47 | 0.48 |
| $derm.$ | 0.04 | **0.07** | 0.04 | 0.03 | 0.02 | 0.04 | 0.06 | 0.05 | 0.03 | 0.03 |
| $wine$ | 0.14 | **0.15** | 0.14 | 0.14 | 0.14 | 0.14 | 0.13 | 0.14 | 0.14 | 0.14 |
| $mm$ | 0.55 | 0.57 | 0.57 | 0.59 | **0.62** | 0.48 | 0.56 | 0.60 | 0.61 | **0.62** |
| $revi.$ | 0.38 | 0.46 | 0.46 | 0.49 | 0.46 | 0.33 | 0.49 | 0.52 | 0.48 | **0.53** |
| $la12$ | 0.35 | 0.39 | 0.36 | 0.32 | 0.38 | 0.33 | 0.36 | 0.33 | 0.35 | **0.41** |
| $spor.$ | 0.32 | 0.40 | 0.34 | 0.37 | 0.33 | 0.28 | **0.41** | 0.34 | 0.30 | 0.31 |
| score | 8.49 | 10.31 | 8.91 | 8.72 | 8.60 | 7.82 | **10.36** | 9.13 | 8.62 | 8.88 |

## 5.1 Experimental Setup

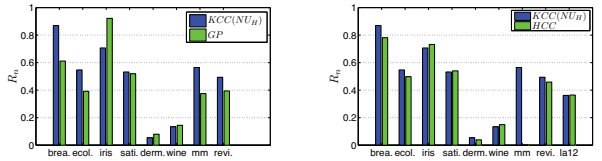
*Validation measure.* Having the class labels, we adopted *normalized Rand index* ($R_n$), a long-standing and robust measure [Jain and Dubes, 1988], for cluster validity.

*Clustering tools.* Three consensus clustering methods, namely KCC, the graph partitioning algorithm (GP), and the hierarchical algorithm (HCC), were employed in the experiments. GP is a tool consisting of three benchmark algorithms: CSPA, HGPA and MCLA [Strehl and Ghosh, 2002]. HCC was implemented by ourselves in MATLAB following the algorithmic description in [Fred and Jain, 2005]. We also implemented KCC in MATLAB, which includes ten utility functions, namely $U_c$, $U_H$, $U_{\cos}$, $U_{L_5}$ and $U_{L_8}$, and their corresponding normalized versions (denoted as $NU_x$).

*Parameter settings.* To generate basic partitionings (BPs), we used the *kmeans* function of MATLAB with squared Euclidean distance for UCI data sets and with cosine similarity for text data sets. The default settings are as follows: 1) $r = 100$; 2) the number of clusters $K$ is set to the number of true clusters; 3) Random Parameter Selection (RPS) is the default generation strategy for BPs by randomizing $K_i$ within $[K, \sqrt{n}]$; 4) $w_i = 1/n, \forall\, i$. For each $\Pi$, KCC and GP were run ten times to obtain the average results, whereas HCC was run only once due to its deterministic nature. In each run, KCC called K-means ten times for the best objective values, and GP called CSPA, HGPA and MCLA only once.

## 5.2 Clustering Efficiency of KCC

The primary concern about consensus clustering is the efficiency. Along this line, we compared KCC (using $U_c$) with GP and HCC in terms of execution efficiency. As can be seen from Table 2, KCC was the fastest one on nine out of eleven data sets, even though it called *kmeans* $10 \times 10$ times on each




(a) KCC vs. GP (b) KCC vs. HCC

Figure 1: Comparison of Clustering Quality

data set. For large-scale data sets, such as *satimage* and *reviews*, the advantage of KCC was particularly evident. HCC seems more suitable for small data sets for its $n$-squared complexity. GP consumed much less time than HCC on large-scale data sets, but it suffered from high space complexity which led to memory failures marked as "N/A" in Table 2.

### 5.3 Clustering Quality of KCC

The clustering results of KCC are shown in Table 4. Eight out of ten utility functions performed the best on at least one data set. For some data sets such as *breast_w* the differences were very sharp. This implies that the diversity of utility functions is indeed beneficial. KCC achieves an edge by allowing the flexible choice of utility functions.

We then validated the robustness of each utility function on all data sets. A utility function $U_i$ is scored by $score(U_i) = \sum_j \frac{R_n(U_i,D_j)}{\max_i R_n(U_i,D_j)}$, where $R_n(U_i, D_j)$ is the $R_n$ value of the clustering result by applying $U_i$ on data set $D_j$. The bottom row of Table 4 shows the scores, where the highest score was won by $NU_H$, closely followed by $U_H$, and then $NU_{\cos}$ and $U_{\cos}$. We therefore took $NU_H$ as the default choice for KCC. It is interesting that though being the first utility function proposed in the literature, $U_c$ and its normalized version generally performed the worst among all.

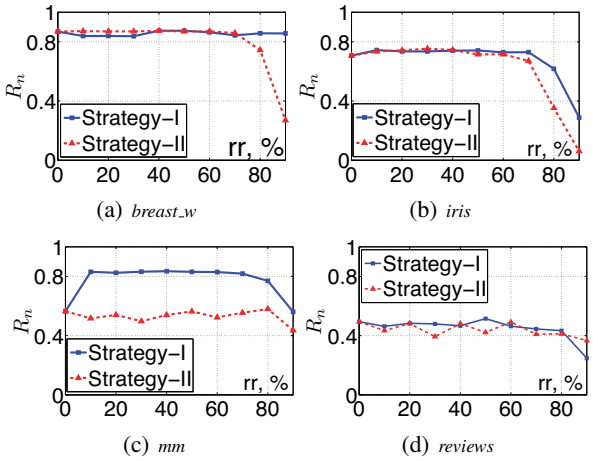
Next we compared KCC using $NU_H$ with the other two methods GP and HCC. As can be seen from Fig. 1, KCC showed comparable clustering performances. Indeed, KCC outperformed GP on 5 out of 8 data sets, and beat HCC on 5 out of 9 data sets. Note that in the left (right) sub-graph we omitted three (two) data sets, on which GP (HCC) suffered memory failures when running on a PC with 4GB RAM.

Note that we so far relied on RPS to generate basic partitionings, which led to poor clustering results on *wine*, *dermatology*, *la12* and *sports*. One way to deal with these is to adjust RPS. Indeed, we found that by narrowing the interval of $K_i$ in RPS to $[2, 2K]$, the performances of KCC on *la12* and *sports* improved substantially. Similar situations happened to *wine* and *dermatology* when we replaced RPS by RFS (Random Feature Selection), where partial data attributes (e.g., 20%) were randomly selected for each basic clustering. We omit details here for space concern.

In summary, KCC is flexible and competitive to GP and HCC in terms of clustering quality. Among the ten utility functions, $NU_H$ shows more robust performances and thus becomes the primary choice for KCC.

### 5.4 Performances on Incomplete BPs

Here, we demonstrate the ability of KCC in handling incomplete BPs (IBP). To this end, we adopted two strategies to




(a) *breast_w* (b) *iris*

(c) *mm* (d) *reviews*

Figure 2: Performances of KCC on Incomplete BPs

generate IBPs. Strategy-I is to randomly remove some data instances from a data set first, and then employ *kmeans* on the incomplete data set to generate an IBP. Strategy-II is to employ *kmeans* on the whole data set first, and then randomly remove some labels from the complete BP. The removal ratio $rr$ was set from 0% to 90% to watch the variations.

Fig. 2 shows that IBPs generally exerted little impact on KCC when $rr < 70\%$. For *mm*, the performance of KCC was even improved from around 0.6 to over 0.8 when $10\% \leq rr \leq 70\%$! This result strongly indicates that KCC is very robust to IBPs — it can recover the whole cluster structure from cluster fragments. It is also interesting to see that the effect of Strategy-I was comparable to the effect of Strategy-II. In some cases, e.g., on *mm*, or on *breast_w* and *iris* when $rr$ is sufficiently large, Strategy-I even led to better performances. This observation is somewhat surprising, since Strategy-II was thought to reserve the structural information more. It is however encouraging since Strategy-I is closer to real-life scenarios of knowledge fusion.

## 6 Conclusions

In this paper, we established a general theoretic framework of K-means-based consensus clustering (KCC). Experiments demonstrated the effectiveness of KCC compared with some state-of-the-art methods. In particular, KCC shows robust performances even if only a few high-quality basic partitionings are available or they suffers severe incompleteness.

## Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (NSFC) under Grants 71171007, 70901002, 71028002 and 71031001, by the Foundation for the Author of National Excellent Doctoral Dissertation of PR China under Grant 201189, and by the Program for New Century Excellent Talents in University under Grant NCET-11-0778. Dr. Jie Cao's work was supported in part by NSFC under Grant 71072172.

## References


[Ayad and Kamel, 2008] H. G. Ayad and M. S. Kamel. Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):160–173, 2008.

[Domeniconi and Al-Razgan, 2009] C. Domeniconi and M. Al-Razgan. Weighted cluster ensembles: Methods and analysis. *ACM Transactions on Knowledge Discovery from Data*, 2(4), 2009.

[Fern and Brodley, 2004] Fern and C. E. Brodley. A survey of clustering ensemble algorithms. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[Filkov and Steven, 2004a] V. Filkov and S. Steven. Heterogeneous data integration with the consensus clustering formalism. *Data Integration in the Life Sciences*, 2004.

[Filkov and Steven, 2004b] V. Filkov and S. Steven. Integrating microarray data by consensus clustering. *International Journal on Artificial Intelligence Tools*, 13(4):863–880, 2004.

[Fischer and Buhmann, 2003] R. Fischer and J. Buhmann. Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4), 2003.

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

[Ghaemi *et al.*, 2009] R. Ghaemi, M. N. Sulaiman, H. Ibrahim, and N. Mustapha. A survey: clustering ensembles techniques. *World Academy of Science, Engineering and Technology*, pages 636–645, 2009.

[Gionis *et al.*, 2007] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data*, 1(1):1–30, 2007.

[Jain and Dubes, 1988] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[Li *et al.*, 2007] T. Li, D. Chris, and I. J. Michael. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Proceedings of 7th IEEE International Conference on Data Mining*, 2007.

[Li *et al.*, 2010] T. Li, M. M. Ogihara, and S. Ma. On combining multiple clusterings: an overview and a new perspective. *Applied Intelligence*, 32(2):207–219, 2010.

[Lu *et al.*, 2008] Z. Lu, Y. Peng, and J. Xiao. From comparing clusterings to combining clusterings. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 361–370, AAAI Press, July 2008. Chicago, Illinois, USA.

[MacQueen, 1967] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, 1967.

[Mirkin, 2001] B. Mirkin. Reinterpreting the category utility function. *Machine Learning*, 45(2):219–228, November 2001.

[Monti *et al.*, 2003] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1-2):91–118, 2003.

[Nguyen and Caruana, 2007] N. Nguyen and R. Caruana. Consensus clusterings. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 607–612, Washington, DC, USA, 2007.

[Strehl and Ghosh, 2002] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[Topchy *et al.*, 2003] A. Topchy, A. Jain, and W. Punch. Combining multiple weak clusterings. In *Proceedings of the 3th IEEE International Conference on Data Mining*, pages 331–338, Melbourne, Florida, USA, November 2003.

[Topchy *et al.*, 2004] A. Topchy, A. Jain, and W. Punch. A mixture model for clustering ensembles. In *Proceedings of the 4th SIAM International Conference on Data Mining*, Florida, USA, 2004.

[Topchy *et al.*, 2005] A. Topchy, A. Jain, and W. Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1866–1881, 2005.

[Vega-Pons and Ruiz-shulcloper, 2011] S. Vega-Pons and J. Ruiz-shulcloper. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(3):337–372, 2011.

[Vega-Pons *et al.*, 2010] S. Vega-Pons, J. Correa-Morris, and J. Ruiz-Shulcloper. Weighted partition consensus via kernels. *Pattern Recognition*, 43(8):2712–2724, 2010.

[Wang *et al.*, 2009] X. Wang, C. Yang, and J. Zhou. Clustering aggregation by probability accumulation. *Pattern Recognition*, 42(5):668–675, 2009.

[Wu *et al.*, 2012] J. Wu, H. Xiong, C. Liu, and J. Chen. A generalization of distance functions for fuzzy $c$-means clustering with centroids of arithmetic means. *IEEE Transactions on Fuzzy Systems*, 20(3):557–571, 2012.

[Yoon *et al.*, 2006] H. S. Yoon, S. Y. Ahn, S. H. Lee, S. B. Cho, and J. Kim. Heterogeneous clustering ensemble method for combining different cluster results. *Data Mining for Biomedical Applications*, pages 82–92, 2006.

[Zhang *et al.*, 2010] Peng Zhang, Xingquan Zhu, Jianlong Tan, and Li Guo. Classifier and cluster ensembling for mining concept drifting data streams. In *Proceedings of the 10th IEEE International Conference on Data Mining*, pages 1175–1180, Sydney, Australia, Dec 2010.