We are IntechOpen, the world’s leading publisher of Open Access books
Built by scientists, for scientists

3,900 Open access books available
116,000 International authors and editors
120M Downloads

154 Countries delivered to
TOP 1% Our authors are among the most cited scientists
12.2% Contributors from top 500 universities

WEB OF SCIENCE™
Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com
Abstract

Clustering is the task of partitioning objects into clusters on the basis of certain criteria so that objects in the same cluster are similar. Many clustering methods have been proposed in a number of decades. Since clustering results depend on criteria and algorithms, appropriate selection of them is an essential problem. Recently, large sets of users’ behavior logs and text documents are common. These are often presented as high-dimensional and sparse vectors. This chapter introduces information-theoretic clustering (ITC), which is appropriate and useful to analyze such a high-dimensional data, from both theoretical and experimental side. Theoretically, the criterion, generative models, and novel algorithms are shown. Experimentally, it shows the effectiveness and usefulness of ITC for text analysis as an important example.

Keywords: information-theoretic clustering, competitive learning, Kullback-Leibler divergence, Jensen-Shannon divergence, clustering algorithm, text analysis

1. Introduction

Clustering is the task of partitioning objects into clusters on the basis of certain criteria so that objects in the same cluster are similar. It is a fundamental procedure to analyze data [1, 2]. Clustering is unsupervised and different from supervised classification. In supervised classification, we have a set of labeled data (belong to predefined classes), train a classifier using the labeled data (training set), and judge which class a new object belongs to by the classifier. In the case of clustering, we find meaningful clusters without using any labeled data and group a given collection of unlabeled data into them. Clustering can also help us to find meaningful classes (labels) for supervised classification. Since it is more difficult to prepare the training set for larger data sets, recently unsupervised analysis of data such as clustering becomes more important.

For example, Table 1 user-item matrix shows which item a user bought. When considering the data as a set of feature vectors for users, we can find a lot of types of users’ behavior by
clustering. It is also possible to analyze data as a set of feature vectors for items. From word-document matrix in Table 2, both document clusters and word clusters could be extracted.

Many clustering methods have been proposed in a number of decades. Those include k-means algorithm [3], competitive learning [4], spherical clustering [5], spectral clustering [6], and maximum margin clustering [7]. Since clustering results depend on criteria and algorithms, appropriate selection of them is an essential problem. Large sets of users’ behavior logs and text documents (as shown in Tables 1 and 2) are common recently. These are often presented as high-dimensional and sparse vectors. This chapter introduces information-theoretic clustering [8] and algorithms that are appropriate and useful to analyze such a high-dimensional data.

Information-theoretic clustering (ITC) uses Kullback-Leibler divergence and Jensen-Shannon divergence to determine its criterion, while k-means algorithm uses the sum of squared error as criterion. This chapter explains ITC by contrasting these two clustering techniques (criteria and algorithms), because there are a number of interesting similarities between them. There exists difficulty in algorithms for ITC. We explain the details of it and propose novel algorithms to overcome.

Experimental results for text data sets are presented to show the effectiveness and usefulness of ITC and novel algorithms for it. In experiments, maximum margin clustering and spherical clustering are used to compare. We also provide the evidence to support the effectiveness of ITC by detailed analysis of clustering results.

2. The sum-of-squared-error criterion and algorithms

Given a set of $M$-dimensional input vectors $X = \{x^i | x^i \in \mathbb{R}^M, i = 1,...,N\}$ where $N$ is the number of vectors, clustering is the task of assigning each input vector $x^i$ a cluster label $k(k = 1,...,K)$ to
partition them into $K$ clusters $C = \{C_1, ..., C_K\}$. The sum-of-squared-error criterion [9] is a simple and widely used criterion for clustering.

Let $\mu^k$ be the mean of the input vectors $x^i$ which belong to the cluster $C_k$ (see Figure 1). Then, the error in $C_k$ is the sum of squared lengths of the differential ("error") vectors $\|x^i - \mu^k\|^2$ and the sum-of-squared-error criterion about all clusters (within-cluster sum of squares) is defined by

$$J_W = \sum_{k=1}^{K} \sum_{x^i \in C_k} \|x^i - \mu^k\|^2.$$  \hspace{1cm} (1)

$J_W$ is the objective function (criterion) to be minimized in clustering based on this criterion.

Also, we define the sum of squares of between-cluster $J_B$ and total $J_T$ as

$$J_B = \sum_{k=1}^{K} N_k \|\mu^k - \mu\|^2,$$

$$J_T = \sum_{i=1}^{N} \|x^i - \mu\|^2,$$  \hspace{1cm} (2)

respectively, where $N_i$ is the number of input vectors $x^i$ in $C_k$ (i.e., $N = \sum_{k=1}^{K} N_k$) and

$$\mu^k = \frac{1}{N_k} \sum_{x^i \in C_k} x^i, \quad \mu = \frac{1}{N} \sum_{i=1}^{N} x^i.$$  \hspace{1cm} (3)

It follows from these definitions that the total sum of squares is the sum of the within-cluster sum of squares and the between-cluster sum of squares:

$$J_T = J_W + J_B.$$  \hspace{1cm} (4)

Since the mean of the all input vectors $\mu$ is derived from $X = \{x^1, ..., x^N\}$ [see Eq. (3)], $J_T$ does not depend on clusters $C$ [see Eq. (2)] and is constant for the given input vectors $X$. Therefore, minimization of $J_W$ is equivalent to maximization of $J_B$. In this sense, clustering based on minimizing this criterion $J_W$ works to find separable clusters each other.

2.1. Generative model

In the background of the clustering based on the objective function (criterion) $J_W$, there exists assumption of Gaussian distribution about input vectors [10].
Suppose that there are clusters \( C^k(k = 1, \ldots, K) \), which generates input vectors by the conditional probability density function:

\[
p(x'|x' \in C^k) = \frac{1}{(2\pi\sigma_k^2)^{M/2}} \exp \left(-\frac{\|x' - \mu_k\|^2}{2\sigma_k^2}\right),
\]

where \( \sigma_k \) is a standard deviation of the cluster \( C^k \) and \( M \) is the number of dimension of \( x' \). In followings, we assume that \( \sigma_k \) is constant value \( \sigma \) for all clusters \( C^k(k = 1, \ldots, K) \). Considering independence of each generation, joint probability density function for the input vectors \( X \) becomes

\[
p(X|C) = \prod_{k=1}^{K} \prod_{x' \in C^k} \frac{1}{(2\pi\sigma^2)^{M/2}} \exp \left(-\frac{\|x' - \mu_k\|^2}{2\sigma^2}\right),
\]

where \( C \) indicate cluster information that specifies which input vector \( x' \) belongs to cluster \( C^k \). Taking the logarithm of Eq. (6) yields

\[
\ln p(X|C) = -\frac{NM}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{k=1}^{K} \sum_{x' \in C^k} \|x' - \mu_k\|^2.
\]

Since \( \sigma \) is constant, the maximization of Eq. (7) is equivalent to the minimization of

\[
\sum_{k=1}^{K} \sum_{x' \in C^k} \|x' - \mu_k\|^2.
\]

which is nothing more or less than the objective function (criterion) \( J_W \). Therefore, under the assumption of Gaussian distribution about input vectors, clustering based on Eq. (8) works to find the most probable solution \( C \).

### 2.2. Algorithms

#### 2.2.1. k-means algorithm

**k-means** [3, 11] is well-known algorithm for clustering based on the sum-of-squared-error criterion. Main idea of this algorithm is as follows. In the objective function \( J_W (1) \), error for vector \( x \) is calculated by \( \|x - \mu_k\|^2 \) where \( \mu_k \) is the mean of cluster \( C^k \) to which \( x \) belongs. If \( \|x - \mu_k\|^2 < \|x - \mu_t\|^2 \), changing the cluster from \( C^k \) to \( C^t \) can reduce the objective function \( J_W \).

We introduce **weight vector** \( w^k(k = 1, \ldots, K) \) that represent cluster \( C^k \) to implement the idea mentioned above. The weight vector \( w^k \) involves mean vector \( \mu_k \) and prototype vector of cluster \( C^k \). As illustrated in Figure 2, the idea of **k-means** is alternative repetition of two steps

(a) Update weights” (calculating mean \( \mu_k \) as weight vector \( w^k \)) and “(b) Update clusters” (allocating input vector \( x' \) to a cluster \( C^k \) on the basis of minimum length from weight vectors \( w^k \)). Note that Figure 2b is a Voronoi tessellation determined by weight vectors \( w^k \), which are usually called **prototype vector** in this context.
Figure 3a is a flow chart of k-means algorithm to which processes of initialization and termination are added. As a matter of fact, clustering is closely related to vector quantization. Vector quantization means mapping input vectors to a codebook that is a set of weight vectors (prototype vectors). When using quantization error $E_Q$:

$$E_Q = \sum_{i=1}^{N} \min_k \|x^i - w^k\|^2,$$  \hspace{1cm} (9)

clusters $C$ determined by a local optimal solution of vector quantization $W$ is a local optimal solution of clustering problem [12]. In this sense, clustering can be replaced by vector quantization and vice versa. We can write a flow chart for vector quantization as Figure 3b, but we also find this chart (b) as k-means algorithm. Furthermore, LBG algorithm [13], which is well known for vector quantization, is based on an approach of Lloyd [3] (one of original papers for k-means algorithm). These facts show a close relationship between clustering and vector quantization.

Initialization is important, because k-means algorithm converges to a local optimal solution which depends on an initial condition (a set of weights or clusters). If we initialize weights $W$ by randomly selecting them from input vectors, it may converge to a very bad local optimal solution with high probability. Random labeling that randomly assigns cluster labels $C$ to input vectors may lead to better solutions than random selection of weights. The initialization Random labeling can also be used for charts (b) and (c) in Figure 3 by replacing “Initialize weights” step to “Initialize clusters” and “Update weights” steps. For directly initializing weights, splitting algorithm [13] and k-means ++ [14] were known.

2.2.2. Competitive learning

Competitive learning [4, 11] is a learning method for vector quantization and also utilized for clustering. While k-means algorithm updates all weights $W$ by batch processing, competitive learning updates one weight $w$ at a time to reduce a part of the quantization error $Q_e$ (see Figure 3c) as
1. Select one input vector $x$ randomly from $X$.

2. Decide a winner $w^c$ from $W$ by

$$c = \text{arg min}_k \|x - w^k\|^2 \quad \text{(If there are several candidates, choose the smallest $k$).} \quad (10)$$

3. Update the winner's weight $w^c$ as

$$w^c \leftarrow (1 - \gamma)w^c + \gamma x, \quad (11)$$

where $\gamma$ is a given learning rate (e.g., 0.01–0.1).

Though the winner-take-all update in Step 3 (Figure 4) that reduces partial error $\|x - w^c\|^2$ in steepest direction does not always reduce the total quantization error $E_Q$, repetition of the update can reduce $E_Q$ on the basis of stochastic gradient decent method [15, 16]. For termination condition, maximum number of times of iteration $N_r$ (the number of maximum repetitions) can be used. After termination, the step of deciding clusters $C$ like Figure 2b is required for clustering purpose.

Figure 3. Flow charts of algorithms based on sum-of-squared-error criterion. (a) k-means1, (b) k-means2, and (c) competitive learning.

Figure 4. Update of the winner's weight in competitive learning.
Against natural expectations, competitive learning outperforms k-means without any contrivance in most cases. Furthermore, information obtained in learning process allows us to improve its performance. Splitting rule [12] utilizes the number of each weight \( w \) wins to estimate density around it. As Figure 5a, b shows, higher density of input vectors around makes the weight vector \( w^i \) win more frequently than \( w^b \).

![Figure 5. Density of input vectors around a weight vector.](image)

Splitting rule in competitive learning [12] aims to overcome the problem of discrepancy between distribution of input vectors \( X \) and that of weight vectors \( W \). The discrepancy causes a few weight vectors \( w \) monopolize \( X \) and leads to a solution of very poor quality, but it is impossible to figure out the distribution of input vectors beforehand. Accordingly, this splitting rule distributes weight vectors \( w \) in learning process as

1. One weight vector \( w^1 \) with a variable \( \tau^1 \) is set. \( \tau^1 \) denotes how many times weight vector \( w^1 \) wins and is initialized to 0.
2. Select one input vector \( x \), decide winner \( w^c \), and update the winner’s weight \( w^c \).
3. Add 1 to \( \tau^c \). If \( \tau^c = \theta \) and the current number of weights \( K' \) is less than \( K \), generate a new weight vector \( w \) which is the same as the winner \( w^c \) and clear \( \tau \) of both to 0, where \( \theta \) is the threshold of times for splitting.
4. Repeat 2 and 3 until termination condition is true.

3. Information-theoretic clustering and algorithms

Information-theoretic clustering (ITC) [8] is closely related to works about distributional clustering [17–19] and uses Kullback-Leibler divergence and Jensen-Shannon divergence to determine its criterion. Though there exists difficulty in algorithms and effectiveness for high-dimensional count data (e.g., text data), its definition and properties are similar to those of the sum-of-squared-error criterion. The main contributions of this chapter are to present the technique to overcome the difficulty and effectiveness of ITC.

Let \( X = \{x^i \in \mathbb{R}^M_+ | i = 1, \ldots, N \} \) be a set of \( M \)-dimensional input vectors (\( N \) denote the number of input vectors), where elements of vectors \( x \) are nonnegative real numbers. We define a \( l^1 \)-norm
of input vector \( \hat{t} = (\sum_m |x_m|) \), normalized input vectors \( p' = x'/t \), and an input probability distribution \( P \) whose \( m \)th random variable takes the \( m \)th element of \( P' = (p'_m) \). Let \( \mathcal{P} = \{P_1, ..., P_N\} \) be a set of input distributions (input data). Suppose that we assign each distribution \( P \) a cluster label \( k = 1, ..., K \) to partition them into \( K \) clusters \( C = \{C_1, ..., C_K\} \).

Let \( \bar{P} \) be the distributions on the mean of input data \( P \) which belong to the cluster \( C_k \) (see Figure 6). Then, the generalized Jensen-Shannon (JS) divergence to be minimized in \( \text{Figure 6} \) is defined by

\[
D_{JS}(\{P^i | P \in C_k\}) = \sum_{P \in C_k} \pi P D_{KL}(P \parallel \bar{P}), \quad \bar{P} = \sum_{P \in C_k} \pi P,
\]

where \( N_k \) is the number of distributions \( P \) in cluster \( C_k \) (i.e., \( N = \sum_{k=1}^{K} N_k \) ), \( D_{KL}(P \parallel \bar{P}) \) is the Kullback-Leibler (KL) divergence to the mean distribution \( \bar{P} \) from \( P \), and \( \pi \) is the probability of \( P \) (\( \sum_{P \in C_k} \pi = 1 \)). Here \( P = 1/N_k \). Then, we define within-cluster JS divergence \( JS_w \) which considers all clusters \( C_k (k = 1, ..., K) \) as

\[
JS_w = \frac{1}{K} \sum_{k=1}^{K} N_k \sum_{P \in C_k} D_{KL}(P \parallel \bar{P}),
\]

\[
= \frac{1}{N} \sum_{k=1}^{K} \sum_{P \in C_k} \sum_{m=1}^{M} p'_{m} \log \frac{p'_{m}}{p''_{m}} = \frac{1}{N} \sum_{k=1}^{K} \sum_{P \in C_k} \sum_{m=1}^{M} (p'_{m} \log p''_{m} - p''_{m} \log p'_{m}).
\]

The within-cluster JS divergence \( JS_w \) is the objective function (criterion) of information-theoretic clustering (ITC) to be minimized \([8]\). We also define JS divergence of between-cluster \( JS_B \) and total \( JS_T \) as

\[
JS_B = D_{JS}(\{P^i | k = 1, ..., K\}) = \sum_{k=1}^{K} \pi_k D_{KL}(P^i \parallel \bar{P}), \quad \pi_k = N_k/N,
\]

\[
JS_T = D_{JS}(\{P^i | i = 1, ..., N\}) = \sum_{i=1}^{N} \pi_i D_{KL}(P^i \parallel \bar{P}), \quad \pi_i = 1/N,
\]

where \( \bar{P} = \sum_{i=1}^{N} \pi_i P^i = 1/N \sum_{i=1}^{N} P^i \) is the distribution on the mean of all input data. It follows from these definitions that the total JS divergence is the sum of the within-cluster JS divergence and the between-cluster JS divergence \([8]\):

\[
JS_T = JS_w + JS_B.
\]

Since \( JS_T \) are constant for given input distributions \( \mathcal{P} \), minimization of \( JS_w \) is equivalent to maximization of \( JS_B \). In this sense, clustering based on minimizing this criterion \( JS_w \) works to find separable clusters each other.
The definition and properties of ITC as shown so far are similar to those of the sum-of-squared-error criterion. Those will help us to understand ITC.

3.1. Generative model

In the background of information-theoretic clustering (ITC), there also exists the bag-of-words assumption [20] that disregards the order of words in a document. (Since ITC is not limited for document clustering, “word” is just an example of feature.) It means that features in data are conditionally independent and identically distributed, where the condition is a given probability distribution for an input vector. Based on this assumption, we describe a generative probabilistic model related to ITC and make clear the relationship between the model and the objective function (criterion) $J_S W$.

Let an input vector $x = [x_1, ..., x_m, ..., x_M]$ present a set of the number of observations of $m$th feature. Suppose that there are clusters $C^k (k = 1, ..., K)$ which generates $t^i$ features for a data (= input vector) with the probability distribution $\mathcal{P}^k = [p^k_1, ..., p^k_M]$, and conditional probability of a set of the observation about features in an input vector $x^i$ is expressed by multinomial distribution

$$p(x^i | x^i \in C^k) = A^i \prod_{m=1}^{M} (p^k_m)^{x^i_m}, \quad A^i = \frac{t^i}{x^i_1! \cdot x^i_2! \cdot \cdots \cdot x^i_M!},$$

where $A^i$ is the number of combination of the observation. Assuming independence of each generation, joint probability function for the input vectors $X = \{x^1, ..., x^N\}$ becomes

$$p(X|C) = \prod_{k=1}^{K} \prod_{x^i \in C^k} A^i \prod_{m=1}^{M} (p^k_m)^{x^i_m},$$

where $C$ indicates cluster information that specifies which input vector $x^i$ belongs to cluster $C^k$. Taking the logarithm of Eq. (20) yields

$$\ln p(X|C) = \sum_{i=1}^{N} \log A^i + \sum_{k=1}^{K} \sum_{x^i \in C^k} x^i_m \log p^k_m$$

$$= \sum_{i=1}^{N} \log A^i + \sum_{k=1}^{K} \sum_{p^i \in C^k} t^p \log p^k_m.$$ (22)

This is a generative probabilistic model related to ITC. If we assume that $t^p$ takes constant value $t$ for all input vectors, maximization of the probability (22) as well as minimization of the objective function $J_S W$ (15) come to the minimization of

$$\frac{1}{N} \sum_{k=1}^{K} \sum_{p^i \in C^k} \sum_{m=1}^{M} -p^i_m \log p^k_m = \frac{1}{N} \sum_{k=1}^{K} N_k \sum_{m=1}^{M} -p^k_m \log p^k_m.$$ (23)

for given input distribution $\mathcal{P}$. Here, the relationship $\sum_{p^i \in C^k} p^i_m = N_k p^k_m$ is used. Since $t^p$ may not be constant value $t$, the generative model (22) is not an equivalent model of ITC but the related
model. This difference comes from the fact that the model treats each observation about features equally, while ITC treats each data (input vector) equally. Though the additional assumption \( t = t \) is required, ITC works to find the most probable solution \( C \) in the generative probabilistic model. Furthermore, Eq. (23) is also based on the minimization of entropy in clusters as Eq. (23) shows. Entropy (specifically, Shannon Entropy) is the expected value of the information contained in each message that is an input distribution here. The smaller entropy becomes, and the more compactly a model can explain observations (input distributions). In this sense, the objective function \( JS_W \) (15) presents the goodness of the generative model. The relationship (including difference) between the probabilistic model and the objective function \( JS_W \) is meaningful to improve the model and the objective function in future.

Choice of appropriate model for data is important, when analyzing them. For example, large set of text documents contain many kinds of words and are presented as high-dimensional vectors. Taking extreme diversity of documents’ topics into account, feature vectors of documents are distributed almost uniformly in the vector space. As known by “the curse of dimensionality” [10], most of the volume of a sphere in high-dimensional space is concentrated near the surface, and it becomes not appropriate to choose the model based on Gaussian distribution which concentrates values around the mean. In contrast, ITC on the basis of the multinomial distribution is a reasonable and useful tool to analyze such a high-dimensional count data, because the generative model of ITC is consistent with them.

We introduce weight distribution \( Q^k(k = 1, \ldots, K) \) that represent clusters \( C_k \) and that involves mean distribution \( \mu_k \) and prototype distribution of cluster \( C_k \) in a manner similar to that of the sum-of-squared-error (SSE) criterion (see Section 2.2.1). Figure 7 shows relationships between parameters in generative models. Parameters are generated or estimated by other parameters to maximize probability of generative model. For example, clustering is the task to find the most probable clusters \( C \) for given input vectors \( X \) or input distributions \( P \). In Figure 7b, constructing a classifier is the task to find \( Q \) for given \( P \) and \( C \) (classes in this context) in training process. Then, it estimates \( C \) for unknown \( P \) using the trained \( Q \). The classifier using multinominal distribution is known as multinominal Naive Bayes classifier [21]. As it shows, ITC and Naive Bayes classifier have a close relationship [18].

3.2. Algorithms

There exists difficulty (Appendix A) in algorithms for ITC. We show a novel idea to overcome it.
3.2.1. Competitive learning

When competitive learning decides a winner for an input distribution $P$, it easily faces the difficulty of calculating KL divergence from $P$ to weight distributions $Q$ (see Appendix A). To overcome this difficulty, we present the idea to change an order of steps in competitive learning (CL). As shown in Figure 8b, CL updates all weights (= weight distributions) before deciding winner by

$$Q^k \leftarrow (1-\gamma)Q^k + \gamma P,$$

where $\gamma$ is a learning rate. Since updated weight distributions $Q^k (k = 1, ..., K)$ include all words (features) of input distribution $P$, it is possible to calculate KL divergence $D_{KL}(P||Q^k)$ for all $k$. In following steps, CL decide a winner $Q^c$ from $Q$ by

$$c = \arg \min_k D_{KL}(P||Q^k) \quad (\text{If there are several candidates, choose the smallest } k),$$

and activate winner’s update and discard others. These steps satisfy the CL’s requirement that it partially reduces value of objective function $JS_W$ in steepest direction with the given learning rate $\gamma$. Here, neither approximation nor distortion is added to the criterion of ITC. Note that updates of weight distributions $Q^k$ before activation are provisional (see Figure 8b).

Related work that avoids the difficulty in calculating KL divergence presented skew divergence [22]. The skew divergence is defined as

$$s_\alpha(P,Q) = D_{KL}(P||\alpha Q + (1-\alpha)P),$$

where $\alpha(0 \leq \alpha \leq 1)$ is the mixture ratio of distributions. The skew divergence is exactly the KL divergence at $\alpha = 1$. When $\alpha = 1 - \gamma$, Eq. (26) becomes similar to Eq. (24). Then, we can rewrite the steps in CL above using the skew divergence as

1. Select one input distribution $P$ randomly from $P$.
2. Decide a winner $Q^c$ from $Q$ by
\[ c = \arg\min_k s_k(P, Q^c) \quad \text{(If there are several candidates, choose the smallest k)}, \] (27)

3. Update the winner’s weight distribution \( Q^c \) as

\[ Q^c \leftarrow (1-\gamma)Q^c + \gamma P, \] (28)

where \( \gamma \) is a learning rate and equal to \( 1-\alpha \) (\( \alpha \) is the mixture ratio for \( s_i \)) usually.

Hence, we call this novel algorithm for ITC as “competitive learning using skew divergence” (sdCL). In addition, splitting rule in competitive learning [12] can also be applied to this algorithm.

3.2.2. k-means type algorithm

Dhillon et al. [8] proposed information-theoretic divisive algorithm which is k-means type algorithm with divisive mechanism and uses KL divergence. However, it still remains the
difficulty to use KL divergence directly. In such a situation, we propose to use the skew divergence instead of KL divergence in k-means type algorithm as

1. Initialize clusters \( C \) of input distribution \( \mathcal{P} \) randomly.

2. Update weight distributions \( \mathcal{Q} \) by

\[
Q^c = \frac{1}{N_k} \sum_{P_i \in C_k} P^i.
\]

3. Update each cluster \( c \) of an input distribution \( P^i \) by

\[
c = \arg \min_k s_\alpha(P^i, Q^k) \quad \text{(If there are several candidates, choose the smallest k),}
\]

where mixture ratio \( \alpha (0 \leq \alpha \leq 1) \) for skew divergence \( s_\alpha \) is 0.99 for example.

4. Repeat 2 and 3 until change ratio of objective function \( J_{\mathcal{W}} \) is less than small value (e.g., \( 10^{-8} \)).

The algorithm itself works well to obtain valuable clustering results. Further, if \( \alpha \) is close to 1, skew divergence \( s_\alpha \) becomes a good approximation of KL divergence. Therefore, restart of learning after termination with \( \alpha \) closer to 1, such as 0.999, 0.9999, ..., may lead to better clustering result.

3.2.3. Other algorithms

Slonim and Tishby [23] proposed an agglomerative hierarchical clustering algorithm, which is a hard clustering version of Information Bottleneck algorithm of Tishby et al. [24]. It is similar to the algorithm of Baker and McCallum [18] and merges just two clusters at every step based on the JS divergence of their distributions. A merit of the agglomerative algorithms is not affected by the difficulty of calculating KL divergence, because it just uses JS divergence. However, a merge of clusters at each step optimizes a local criterion but not a global criterion, as Dhillon et al. [8] pointed out. Therefore, clustering results may not be as good as results obtained by nonhierarchical algorithms (e.g., k-means and competitive learning) in the sense of optimizing the objective function of ITC. Additionally, hierarchical algorithms are computationally expensive, when the number of inputs is large.

Note that a lot of studies [8, 18, 23] aimed at improving accuracy of text classification using feature/word clustering based on ITC or distributional clustering. If a clustering is just a step to final goal, feature clustering is meaningful. However, features which characterize clusters should not be merged, when we aim to find clusters (topics) from a set of documents. Actually, finding topics using clustering is the aim of this chapter.

4. Evaluation of clustering

Since clustering results depend on methods (criteria and algorithms), appropriate selection of them is important. So far, we introduced two criteria for clustering. These are called ***internal***
criteria that depend on their own models and not enough for evaluation. If criterion for clustering is common, we can compare clustering results by objective function of the criterion. Under a certain model that is an assumption in other word, a more probable result can be regarded as a better result. However, it is not guaranteed that the model or the assumption is reasonable at all times. Moreover, good clustering results under a certain criterion can be bad results under different criteria. A view from outside is required.

This section introduces external criteria that are Purity, Rand index (RI), and Normalized mutual information (NMI) [25] to evaluate clustering quality and to find better clustering methods. These criteria compare clusters with a set of classes, which are produced on the basis of human judges. Here, each input data belong to one of class \( A^j \) \((j=1, \ldots, J)\) and one of cluster \( C^k \) \((k=1, \ldots, K)\). Let \( T(C^k, A^j) \) be the number of data that belongs to both \( C^k \) and \( A^j \).

Purity is measured by counting the number of input data from the most frequent class in each cluster. Purity can be computed as

\[
purity = \frac{1}{N} \sum_{k=1}^{K} \max_j T(C^k, A^j),
\]

where \( N \) is the total number of input data. Purity is close to 1, when each cluster has one dominant class.

Rand index (RI) checks all of the \( N(N-1)/2 \) pairs of input data and is defined by

\[
RI = \frac{a + b}{a + b + c + d},
\]

where \( a, b, c, \) and \( d \) are the number of pairs in following conditions:

- “a,” where the cluster number (suffix) is the same and the class number is the same
- “b,” where the cluster numbers are different and the class numbers are different
- “c,” where the cluster number is the same and the class numbers are different
- “d,” where the cluster numbers are different and the class number is the same

The Rand index (RI) measures the percentage of agreements \( a+b \) in clusters and classes.

Normalized mutual information (NMI) is defined as

\[
NMI = \frac{I(C; A)}{\left( H(C) + H(A) \right)/2},
\]

where, \( I(C; A) \) is mutual information and \( H(\cdot) \) is entropy and
\[
I(C;A) = \sum_{k=1}^{K} \sum_{j=1}^{J} P(C^k,A^j) \log \frac{P(C^k,A^j)}{P(C^k)P(A^j)} = \sum_{k=1}^{K} \sum_{j=1}^{J} \frac{T(C^k,A^j)N}{T(C^k)T(A^j)}.
\]

where \(P(C^k), P(A^j),\) and \(P(C^k,A^j)\) are the probability of data being in cluster \(C^k\), class \(A^j\), and in the intersection of \(C^k\) and \(A^j\), respectively. Mutual information \(I(C;A)\) measures the mutual dependence between clusters \(C\) and classes \(A\). It quantifies the amount of information obtained for classes through knowing about clusters. Hence, high NMI shows some kind of goodness about clustering in information theory.

5. Experiments

This section provides experimental results that show the effectiveness and usefulness of ITC and the proposed algorithm (sdCL: competitive learning using skew divergence). Experiments consist of two parts, experiment1 and experiment2.

In experiment1, we applied sdCL to the same data sets as used in the paper of Wang et al. [26] and compared performance of sdCL with other clustering algorithms evaluated in it. The algorithms that the paper [26] evaluated are as follows.

- **k-means (KM).** The weights \(W\) are initialized by randomly [26].
- **Normalized cut (NC)** [27].
- **Maximum margin clustering (MMC)** [7].
- **Generalized maximum margin clustering (GMMC)** [28].
- **Iterative support vector regression (IterSVR)** [29].
- **Cutting plane maximum margin clustering (CPMMC)** [26], **CPM3C** [26].

As shown above, maximum margin clustering (MMC) [7] and related works are much focused. These works extend the idea of support vector machine (SVM) [30] to the unsupervised scenario. The experimental results obtained by the MMC technique are often better than conventional clustering methods. Among those, CPMMC and CPM3C (Cutting plane multiclass maximum margin clustering) [26] are known as successful methods. Experimental
results will show that the proposed algorithm sdCL outperforms CPM3C in text data clustering.

In experiment 2, we focus on text data clustering and compare performance of algorithms, sdCL, sdCLS (sdCL with splitting rule, see Sections 2.2.2 and 3.2.1), and spherical competitive learning (spCL). We also provide the evidence to support the effectiveness of ITC by detailed analysis of clustering results.

spCL is an algorithm for spherical clustering like the spherical k-means algorithm [5] that was proposed for clustering high-dimensional and sparse data, such as text data. The objective function to be maximized for the spherical clustering is cosine similarity between input vectors and the mean vector of a cluster to which they belong. To implement spCL, we turn input and weight vectors \( (x, w) \) into a unit vector and decide winner \( w^c \) by

\[
c = \arg \max_k \cos(x, w^k) \quad \text{(If there are several candidates, choose the smallest \( k \))}
\]

and update the winner's weight \( w^c \) as

\[
w^c \leftarrow \frac{(1-\gamma)w^c + \gamma x}{\| (1-\gamma)w^c + \gamma x \|}.
\]

For all competitive learning algorithms, the learning rate \( \gamma = 0.01 \), the number of maximum repetitions for updating weights \( N_r = 1,000,000 \) (termination condition), and the threshold of times for splitting rule \( \theta = 1000 \) are used. After competitive learning (sdCL, sdCLS, or spCL) is terminated, we apply k-means type algorithm to remove fluctuation as a post-processing. Specifically, sdKM (the k-means type algorithm using skew divergence shown in Section 3.2.2) with \( \alpha = 0.999, 0.9999, 0.99999 \) is applied consecutively after sdCL and sdCLS. In each learning procedure including post-processing, an operation is iterated 50 times with different initial random seeds for a given set of parameters.

5.1. Data sets

We mainly use the same data sets as used in the paper of Wang et al. [26]. When applying algorithms for ITC, we use probability distributions \( P_i(i=1, \ldots, N) \) \((P)\) derived from original data.

1. UCI data. From the UCI repository,\(^7\) we use ionosphere, digits, letter, and satellite under the same setting of the paper [26]. The digits data \((8 \times 8 \text{ matrix})\) are generated from bitmaps of handwritten digits. Pairs (3 vs. 8, 1 vs. 7, 2 vs. 7, and 8 vs. 9) are focused due to the difficulty of differentiating. For the letter and satellite data sets, their first two classes are used. Since the ionosphere data contain minus values and cannot be transformed to probability distributions, we do not apply ITC to them.

2. **Text data.** Four text data sets: 20Newsgroups (http://qwone.com/~jason/20Newsgroups/ [Accessed: 2016-10-25]), WebKB,\(^2\) Cora [31], and RCV1 (Reuters Corpus Volume 1) [32] are used. In experiment1, we follow the setting of the paper [26]. For 20Newsgroups data set, topic \(\text{“rec”} \) which contains four topics \{autos, motorcycles, baseball, hockey\} is used. From the four topics, two sets of two-class data sets \{Text-1: autos vs motorcycles, Text-2: baseball vs hockey\} are extracted. From WebKB data sets, the four Universities data set (Cornell, Texas, Washington, and Wisconsin University), which has seven classes (student, faculty, staff, department, course, project, and other), are used. Note that topic of the “other” class is ambiguous and may contain various topics (e.g., faculty), because it is a collection of pages that were not deemed the “main page” representing an instance of the other six classes, as pointed out in the web page of the data set. Cora data set (Cora research paper classification) [31] is a set of information of research papers classified into a topic hierarchy. From this data set, papers in subfield \{data structure (DS), hardware and architecture (HA), machine learning (ML), operating system (OS), programming language (PL)\} are used. We select papers that contain title and abstract. RCV1 data set contains more than 800 thousands documents to which topic category is assigned. The documents with the highest four topic codes (CCAT, ECAT, GCAT, and MCAT) in the topic codes hierarchy in the training set. Multi-labeled instances are removed.

In experiment2, we use all of 20Newsgroups and RCV1 data sets. For RCV1 data set, we obtain 53 classes (categories) by mapping the data set to the second level of RCV1 topic hierarchy and remove multi-labeled instances. For WebKB data set, we remove “other” class due to ambiguity, use the other six classes, and do not use information of universities.

For all text data, we remove \textit{stop words} using \textit{stop list} [32] and empty data, if they are not removed. In experiment1, we follow the setting of the paper [26], but properties of data sets are slightly different (see Table 3). For Cora data sets, the differences of data sizes are large. However, they must keep the same (or close at least) characteristics (e.g., distributions of words and topics), because they are extracted from the same source.

3. **Digits data.** USPS \((16 \times 16)\) and MNIST \((28 \times 28)\) are data sets of handwritten digits image.\(^4\) For USPS data set, 1, 2, 3, and 4 digit images are used. For MNIST and digits data from UCI repository, all 45 pairs of digits 0–9 are used in two-class problems.

The properties of those data sets are listed in Table 3.

5.2. Results of experiment1

The clustering results are shown in Tables 4–7, where values (except for sdCL) are the same in the paper of Wang et al. [26] (accuracy in that paper is equivalent to purity from its definition). In two-class problems, CPMMC outperforms other algorithms about purity and Rand Index (RI) in most cases. The proposed algorithm sdCL shows stable performances except for

\(^1\)http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/ [Accessed: 2016-10-25].
ionosphere to which sdCL cannot be applied. In multiclass problems, sdCL for text data (Cora, 20Newsgroups-4, and Reuters-RCV1-4) outperforms other algorithms. The results show that ITC and the proposed algorithm sdCL are effective for text data sets. Note that CPM3C shows the better results than sdCL for WebKB data. However, topic of the “other” class in WebKB is ambiguous (see Section 5.1). The occupation ratio of them is large [0.710, 0.689, 0.777, 0.739] and almost same as the values of purity in CPM3C and sdCL. It means that these algorithms failed to find meaningful clusters in purity. Therefore, WebKB data are not appropriate to use for evaluation without removing “other” class.

5.3. Results of experiment 2

In experiment 2, we focus on text data clustering. Table 8 shows that the proposed algorithms for ITC (sdCL and sdCLS) outperform spCL in purity, RI, and NMI. Considering that spCL is an algorithm for spherical clustering [5] which was proposed to analyze high-dimensional

<table>
<thead>
<tr>
<th>Data</th>
<th>Size (N)</th>
<th>Feature (M)</th>
<th>Class (K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ionosphere</td>
<td>351</td>
<td>34</td>
<td>2</td>
</tr>
<tr>
<td>Letter</td>
<td>1555</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Digits</td>
<td>1555</td>
<td>64</td>
<td>2</td>
</tr>
<tr>
<td>Satellite</td>
<td>2236</td>
<td>36</td>
<td>2</td>
</tr>
<tr>
<td>Text-1</td>
<td>1981</td>
<td>16,259</td>
<td>2</td>
</tr>
<tr>
<td>Text-2</td>
<td>1987</td>
<td>15,955</td>
<td>2</td>
</tr>
<tr>
<td>20Newsgroups-4</td>
<td>3967</td>
<td>24,506</td>
<td>4</td>
</tr>
<tr>
<td>20Newsgroups-20</td>
<td>18,772</td>
<td>60,698</td>
<td>20</td>
</tr>
<tr>
<td>Cora-DS</td>
<td>2397</td>
<td>5745</td>
<td>9</td>
</tr>
<tr>
<td>Cora-HA</td>
<td>913</td>
<td>3340</td>
<td>7</td>
</tr>
<tr>
<td>Cora-ML</td>
<td>3569</td>
<td>6809</td>
<td>7</td>
</tr>
<tr>
<td>Cora-OS</td>
<td>2084</td>
<td>5029</td>
<td>4</td>
</tr>
<tr>
<td>Cora-PL</td>
<td>3026</td>
<td>6069</td>
<td>9</td>
</tr>
<tr>
<td>WebKB-Cornell</td>
<td>835</td>
<td>5574</td>
<td>7</td>
</tr>
<tr>
<td>WebKB-Texas</td>
<td>808</td>
<td>4482</td>
<td>7</td>
</tr>
<tr>
<td>WebKB-Washington</td>
<td>1191</td>
<td>7779</td>
<td>7</td>
</tr>
<tr>
<td>WebKB-Wisconsin</td>
<td>1218</td>
<td>8270</td>
<td>7</td>
</tr>
<tr>
<td>WebKB6</td>
<td>4219</td>
<td>14,142</td>
<td>6</td>
</tr>
<tr>
<td>Reuters-RCV1-4</td>
<td>19,806</td>
<td>44,214</td>
<td>4</td>
</tr>
<tr>
<td>Reuters-RCV1-53</td>
<td>534,135</td>
<td>216,704</td>
<td>53</td>
</tr>
<tr>
<td>MNIST</td>
<td>70,000</td>
<td>784</td>
<td>2</td>
</tr>
<tr>
<td>USPS</td>
<td>3046</td>
<td>256</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 3. Properties of data sets.
Table 4. Purity comparisons for two-class problems.

<table>
<thead>
<tr>
<th>Data</th>
<th>KM</th>
<th>NC</th>
<th>MMC</th>
<th>GMMC</th>
<th>IterSVR</th>
<th>CPMMC</th>
<th>sdCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iono</td>
<td>0.56</td>
<td>0.63</td>
<td>0.67</td>
<td>0.64</td>
<td>0.56</td>
<td>0.65</td>
<td>–</td>
</tr>
<tr>
<td>Letter</td>
<td>0.71</td>
<td>0.64</td>
<td>–</td>
<td>–</td>
<td>0.87</td>
<td>0.92</td>
<td>0.86</td>
</tr>
<tr>
<td>Satellite</td>
<td>0.92</td>
<td>0.92</td>
<td>–</td>
<td>–</td>
<td>0.94</td>
<td>0.97</td>
<td>0.91</td>
</tr>
<tr>
<td>Text-1</td>
<td>0.50</td>
<td>0.88</td>
<td>–</td>
<td>–</td>
<td>0.94</td>
<td>0.94</td>
<td>0.87</td>
</tr>
<tr>
<td>Text-2</td>
<td>0.50</td>
<td>0.84</td>
<td>–</td>
<td>–</td>
<td>0.89</td>
<td>0.93</td>
<td>0.92</td>
</tr>
<tr>
<td>Dig 3-8</td>
<td>0.90</td>
<td>0.55</td>
<td>0.82</td>
<td>0.90</td>
<td>0.94</td>
<td>0.95</td>
<td>0.89</td>
</tr>
<tr>
<td>Dig 1-7</td>
<td>0.99</td>
<td>0.50</td>
<td>0.57</td>
<td>0.96</td>
<td>0.99</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Dig 2-7</td>
<td>0.94</td>
<td>0.53</td>
<td>0.98</td>
<td>0.99</td>
<td>1.0</td>
<td>1.0</td>
<td>0.98</td>
</tr>
<tr>
<td>Dig 8-9</td>
<td>0.84</td>
<td>0.50</td>
<td>0.93</td>
<td>0.73</td>
<td>0.93</td>
<td>0.97</td>
<td>0.81</td>
</tr>
<tr>
<td>UCI-dig</td>
<td>0.93</td>
<td>0.96</td>
<td>–</td>
<td>–</td>
<td>0.97</td>
<td>0.99</td>
<td>0.93</td>
</tr>
<tr>
<td>MNIST</td>
<td>0.81</td>
<td>0.82</td>
<td>–</td>
<td>–</td>
<td>0.86</td>
<td>0.94</td>
<td>0.83</td>
</tr>
</tbody>
</table>

Bold fonts indicate the maximum purities for a given data set.

Table 5. Rand Index (RI) comparisons for two-class problems.

<table>
<thead>
<tr>
<th>Data</th>
<th>KM</th>
<th>NC</th>
<th>MMC</th>
<th>GMMC</th>
<th>IterSVR</th>
<th>CPMMC</th>
<th>sdCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iono</td>
<td>0.5428</td>
<td>0.7500</td>
<td>0.7875</td>
<td>0.7160</td>
<td>0.7770</td>
<td>0.7548</td>
<td>–</td>
</tr>
<tr>
<td>Letter</td>
<td>0.8206</td>
<td>0.7680</td>
<td>–</td>
<td>–</td>
<td>0.9280</td>
<td>0.9502</td>
<td>0.9267</td>
</tr>
<tr>
<td>Satellite</td>
<td>0.9593</td>
<td>0.9579</td>
<td>–</td>
<td>–</td>
<td>0.9682</td>
<td>0.9879</td>
<td>0.9506</td>
</tr>
<tr>
<td>Text-1</td>
<td>0.5053</td>
<td>0.9379</td>
<td>–</td>
<td>–</td>
<td>0.9702</td>
<td>0.9500</td>
<td>0.9306</td>
</tr>
<tr>
<td>Text-2</td>
<td>0.5038</td>
<td>0.9135</td>
<td>–</td>
<td>–</td>
<td>0.9399</td>
<td>0.9721</td>
<td>0.9591</td>
</tr>
<tr>
<td>Dig 3-8</td>
<td>0.9468</td>
<td>0.6500</td>
<td>0.9000</td>
<td>0.9440</td>
<td>0.9664</td>
<td>0.9688</td>
<td>0.9440</td>
</tr>
<tr>
<td>Dig 1-7</td>
<td>0.9445</td>
<td>0.5500</td>
<td>0.6875</td>
<td>0.9780</td>
<td>0.9945</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Dig 2-7</td>
<td>0.9691</td>
<td>0.6600</td>
<td>0.9875</td>
<td>0.9950</td>
<td>1.000</td>
<td>1.000</td>
<td>0.9891</td>
</tr>
<tr>
<td>Dig 8-9</td>
<td>0.9068</td>
<td>0.5200</td>
<td>0.9625</td>
<td>0.8400</td>
<td>0.9633</td>
<td>0.9812</td>
<td>0.8910</td>
</tr>
<tr>
<td>UCI-dig</td>
<td>0.9638</td>
<td>0.9757</td>
<td>–</td>
<td>–</td>
<td>0.9818</td>
<td>0.9940</td>
<td>0.9516</td>
</tr>
<tr>
<td>MNIST</td>
<td>0.8921</td>
<td>0.8992</td>
<td>–</td>
<td>–</td>
<td>0.9241</td>
<td>0.9621</td>
<td>0.8812</td>
</tr>
</tbody>
</table>

Bold fonts indicate the maximum Rand indices for a given data set.

Table 8 also shows that sdCLS (sdCL with splitting rule, see Sections 2.2.2 and 3.2.1) is slightly better than sdCL in some cases. As far as Figure 9 (left, right) shows, values of JS divergence for sdCLS are smaller (“better” in ITC) than sdCL, and sdCLS outperforms sdCL in purity on average. Nevertheless, an advantage of sdCLS against sdCL is not so obvious in this experiment.

data such as text documents, the criterion of information-theoretic clustering is worth to use for this purpose.
**Table 6.** Purity comparisons for multiclass problems.

<table>
<thead>
<tr>
<th>Data</th>
<th>KM</th>
<th>NC</th>
<th>MMC</th>
<th>CPM3C</th>
<th>sdCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCI-digits 0689</td>
<td>0.696</td>
<td>0.939</td>
<td>0.941</td>
<td>0.974</td>
<td>0.945</td>
</tr>
<tr>
<td>UCI-digits 1279</td>
<td>0.4042</td>
<td>0.9011</td>
<td>0.9191</td>
<td>0.945</td>
<td>0.868</td>
</tr>
<tr>
<td>USPS</td>
<td>0.932</td>
<td>0.938</td>
<td>–</td>
<td>0.950</td>
<td>0.958</td>
</tr>
<tr>
<td>Cora-DS</td>
<td>0.589</td>
<td>0.744</td>
<td>–</td>
<td>0.746</td>
<td>0.823</td>
</tr>
<tr>
<td>Cora-HA</td>
<td>0.385</td>
<td>0.659</td>
<td>–</td>
<td>0.695</td>
<td>0.767</td>
</tr>
<tr>
<td>Cora-ML</td>
<td>0.514</td>
<td>0.720</td>
<td>–</td>
<td>0.761</td>
<td>0.802</td>
</tr>
<tr>
<td>Cora-OS</td>
<td>0.518</td>
<td>0.522</td>
<td>–</td>
<td>0.730</td>
<td>0.735</td>
</tr>
<tr>
<td>Cora-PL</td>
<td>0.643</td>
<td>0.675</td>
<td>–</td>
<td>0.712</td>
<td>0.819</td>
</tr>
<tr>
<td>WebKB-Cornell</td>
<td>0.603</td>
<td>0.602</td>
<td>–</td>
<td>0.724</td>
<td>0.483</td>
</tr>
<tr>
<td>WebKB-Texas</td>
<td>0.604</td>
<td>0.602</td>
<td>–</td>
<td>0.712</td>
<td>0.495</td>
</tr>
<tr>
<td>WebKB-Washington</td>
<td>0.616</td>
<td>0.581</td>
<td>–</td>
<td>0.752</td>
<td>0.426</td>
</tr>
<tr>
<td>WebKB-Wisconsin</td>
<td>0.581</td>
<td>0.509</td>
<td>–</td>
<td>0.761</td>
<td>0.464</td>
</tr>
<tr>
<td>20Newsgrroups-4</td>
<td>0.581</td>
<td>0.496</td>
<td>–</td>
<td>0.780</td>
<td>0.940</td>
</tr>
<tr>
<td>Reuters-RCV1-4</td>
<td>0.471</td>
<td>–</td>
<td>–</td>
<td>0.703</td>
<td>0.800</td>
</tr>
</tbody>
</table>

Bold fonts indicate the maximum purities for a give data set.
Note that clustering result by dCL (competitive learning using KL divergence) is shown below. Since the values of NMI clearly illustrate that dCL converged to unrelated solutions to the classes, use of skew divergence is an effective technique to overcome this problem.

<table>
<thead>
<tr>
<th>Data</th>
<th>Purity</th>
<th>RI</th>
<th>NMI</th>
<th>Purity</th>
<th>RI</th>
<th>NMI</th>
<th>Purity</th>
<th>RI</th>
<th>NMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cora-DS</td>
<td>0.506</td>
<td>0.823</td>
<td>0.322</td>
<td>0.514</td>
<td>0.327</td>
<td>0.361</td>
<td>0.796</td>
<td>0.162</td>
<td></td>
</tr>
<tr>
<td>Cora-HA</td>
<td>0.614</td>
<td>0.767</td>
<td>0.363</td>
<td>0.618</td>
<td>0.360</td>
<td>0.518</td>
<td>0.728</td>
<td>0.251</td>
<td></td>
</tr>
<tr>
<td>Cora-ML</td>
<td>0.597</td>
<td>0.802</td>
<td>0.384</td>
<td>0.602</td>
<td>0.385</td>
<td>0.431</td>
<td>0.757</td>
<td>0.181</td>
<td></td>
</tr>
<tr>
<td>Cora-OS</td>
<td>0.669</td>
<td>0.735</td>
<td>0.313</td>
<td>0.685</td>
<td>0.325</td>
<td>0.582</td>
<td>0.673</td>
<td>0.187</td>
<td></td>
</tr>
<tr>
<td>Cora-PL</td>
<td>0.473</td>
<td>0.819</td>
<td>0.275</td>
<td>0.484</td>
<td>0.274</td>
<td>0.414</td>
<td>0.802</td>
<td>0.175</td>
<td></td>
</tr>
<tr>
<td>WebKB6</td>
<td>0.575</td>
<td>0.712</td>
<td>0.272</td>
<td>0.575</td>
<td>0.272</td>
<td>0.519</td>
<td>0.702</td>
<td>0.181</td>
<td></td>
</tr>
<tr>
<td>20News-20</td>
<td>0.690</td>
<td>0.954</td>
<td>0.653</td>
<td>0.697</td>
<td>0.656</td>
<td>0.359</td>
<td>0.905</td>
<td>0.324</td>
<td></td>
</tr>
<tr>
<td>RCV1-53</td>
<td>0.731</td>
<td>0.910</td>
<td>0.586</td>
<td>0.738</td>
<td>0.580</td>
<td>0.568</td>
<td>0.895</td>
<td>0.384</td>
<td></td>
</tr>
</tbody>
</table>

Table 8. Comparison for text data sets.

Figure 9. Purity versus JS divergence for 20Newsgroups (left) and Reuters-RCV1 (right) data sets.

Note that clustering result by dCL (competitive learning using KL divergence) is shown below. Since the values of NMI clearly illustrate that dCL converged to unrelated solutions to the classes, use of skew divergence is an effective technique to overcome this problem.

<table>
<thead>
<tr>
<th>Data</th>
<th>Purity</th>
<th>RI</th>
<th>NMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebKB6</td>
<td>0.363</td>
<td>0.667</td>
<td>0.00629</td>
</tr>
<tr>
<td>20News-20</td>
<td>0.071</td>
<td>0.905</td>
<td>0.00596</td>
</tr>
</tbody>
</table>
In followings, we examine inside of clustering results obtained by ITC to make clear whether ITC helps us to find meaningful clusters and candidates of classes for classification. Tables 9 and 10 show frequent words in classes and clusters obtained by sdCL of 20Newsgroups data set, respectively. The order of clusters is arranged so that clusters are made to correspond to classes. Table 11 is the cross table between clusters and classes. As shown in Table 10, the frequent words in some clusters remind us characteristics of them to distinguish from others. For example, the words in cluster 2: “image graphics jpeg,” cluster 6: “sale offer shipping,” and cluster 11: “key encryption chip” remind classes (comp.graphics), (misc.forsale), and (sci.crypt), respectively. We also imagine characteristics of clusters from the words in 7, 8, 9, 10, 13, 14, and 16th clusters. These clusters have documents of one dominant class and can be regarded as candidates of classes. However, there are some exceptions. The 1st and 15th clusters have the same word “god,” while classes of (alt.atheism), (soc.religion.christian), and (talk.religion.misc) have also the same word “god.” The cluster 1 and class (alt.atheism) have common words “religion evidence,” and the cluster 1 has many documents of the dominant class (alt.atheism). The cluster 15 and the class (soc.religion.christian) have common words “jesus bible christ church,” and the cluster 15 has many documents of the dominant class (soc.religion.christian). On the other hand, there is no cluster which has

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Frequent Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>image graphics jpeg, bit images, software data files, ftp</td>
</tr>
<tr>
<td>6</td>
<td>sale, offer, shipping, mail, price, drive, condition, dos, st, email</td>
</tr>
<tr>
<td>11</td>
<td>key, encryption, chip, government, chip, writes, clipper, people, article, keys, system</td>
</tr>
</tbody>
</table>

Table 9. Frequent words in classes of 20Newsgroups data set.
documents of (talk.religion.misc) as dominant, and the documents of (talk.religion.misc) are mostly shared by the clusters 1 and 15. Though there is a mismatch between clusters and classes, the clustering result is also acceptable, because words in the class (talk.religion.misc) are resemble those in the class (soc.religion.christian). We can also find that cluster 4 has many documents of the two classes (comp.sys.ibm.pc.hardware) and (comp.sys.mac.hardware). From Table 9, those classes have similar words except for “mac” and “apple.” Thus, ITC missed to detect the difference of the classes, but found the cluster with common feature of them. In this sense, the clustering result is meaningful and useful. Another example is that documents in class (talk.politics.mideast) are divided into clusters 17 and 18. It means that ITC found two topics from one class and frequency words in the clusters seem to be reasonable (see 17th and 18th clusters in Table 10). The characteristic of cluster 20 that has words “mail list address email send information ca internet article writes” is different from all classes as well as other clusters, but the cluster 20 has some documents in all classes (see Table 11). This cluster may discover that all newsgroups include documents with such words. In summary, ITC helps us to find meaningful clusters, even when clusters obtained by ITC sometimes seem not to be the same as expected classes. The detailed analysis of the clustering results above could be the evidence to support the effectiveness and usefulness of ITC.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>writes god article people religion evidence science moral objective time</td>
</tr>
<tr>
<td>2</td>
<td>image graphics jpeg file images ftp data bit files format</td>
</tr>
<tr>
<td>3</td>
<td>windows dos file system writes os files article program software</td>
</tr>
<tr>
<td>4</td>
<td>drive scsi mb card writes system mac article bit apple</td>
</tr>
<tr>
<td>5</td>
<td>window file server program motif sun widget set display output</td>
</tr>
<tr>
<td>6</td>
<td>sale offer shipping mail price condition st good email interested</td>
</tr>
<tr>
<td>7</td>
<td>car writes article cars engine good ve apr time oil</td>
</tr>
<tr>
<td>8</td>
<td>writes article bike dod apr ca ride ve good time</td>
</tr>
<tr>
<td>9</td>
<td>writes year article game team baseball good games time hit</td>
</tr>
<tr>
<td>10</td>
<td>game team hockey writes play ca games article year season</td>
</tr>
<tr>
<td>11</td>
<td>key encryption government chip writes clipper people article keys system</td>
</tr>
<tr>
<td>12</td>
<td>db writes article power good ground ca time ve circuit</td>
</tr>
<tr>
<td>13</td>
<td>writes article medical people disease health cancer patients time msg</td>
</tr>
<tr>
<td>14</td>
<td>space writes nasa article earth launch orbit time shuttle system</td>
</tr>
<tr>
<td>15</td>
<td>god jesus people bible writes christ christian church christians article</td>
</tr>
<tr>
<td>16</td>
<td>people writes gun article government fbi fire guns koresh batf</td>
</tr>
<tr>
<td>17</td>
<td>israel israeli writes jews article people arab jewish arabs state</td>
</tr>
<tr>
<td>18</td>
<td>people armenian turkish armenians president armenia war mr turks turkey</td>
</tr>
<tr>
<td>19</td>
<td>writes people article government cramer apr state make health optilink</td>
</tr>
<tr>
<td>20</td>
<td>mail list address email send information ca internet article writes</td>
</tr>
</tbody>
</table>

Table 10. Frequent words in clusters of 20Newsgroups data set.
6. Conclusion

In this chapter, we introduced information-theoretic clustering (ITC) from both theoretical and experimental side. Theoretically, we have shown the criterion, generative model, and novel algorithms for ITC. Experimentally, we showed the effectiveness and usefulness of ITC for text analysis as an important example.

A Difficulty about KL divergence

Let $P$ and $Q$ be a distribution whose $m$th random variable $p_m$ and $q_m$ takes the $m$th element of a vector $p$ and $q$, respectively. The Kullback-Leibler (KL) divergence to $Q$ from $P$ is defined to be

$$D_{\text{KL}}(P\|Q) = p_m \log \frac{p_m}{q_m}$$

(39)

In this definition, it is assumed that the support set of $P$ is a subset of the support set of $Q$ (If $q_m$ is zero, $p_m$ must be zero). For a given cluster $C_k$, there is no problem to calculate JS divergence.
of cluster $C^k$ by Eq. (12), because the support set of any distribution $P_i(\in C^k)$ is the subset of the mean distribution $P^k$. However, it is not guaranteed that KL divergence from $P_i(\in C^k)$ to $Q_t(t \neq k)$ (a weight distribution of other cluster $C^t$) is finite. This causes a serious problem to find similar weight distribution $Q$ for an input distribution $P$. For example, lack of even one word (feature) in a distribution $Q$ is enough not to be similar. Therefore, it is difficult to use $k$-means type algorithm,\(^5\) which updates weights or clusters by batch processing, in ITC.

**Acknowledgements**

This work was supported by JSPS KAKENHI Grant Number 26330259.

**Author details**

Toshio Uchiyama

Address all correspondence to: uchiyama.toshio@do-johodai.ac.jp

Hokkaido Information University, Ebetsu-shi, Hokkaido, Japan

**References**


\(^5\)k-means algorithm is known as algorithm for clustering based on the sum-of-squared-error criterion (see Section 2.2.1). Therefore we added word “type” for information-theoretic clustering.


