Information Bottleneck Co-clustering

Pu Wang* Carlotta Domeniconi* Kathryn Blackmond Laskey †

Abstract

Co-clustering has emerged as an important approach for mining contingency data matrices. We present a novel approach to co-clustering based on the Information Bottleneck principle, called Information Bottleneck Co-clustering (IBCC), which supports both soft-partition and hard-partition co-clusterings, and leverages an annealing-style strategy to bypass local optima. Existing co-clustering methods require the user to define the number of row- and column-clusters respectively. In practice, though, the number of row- and column-clusters may not be independent. To address this issue, we also present an agglomerative Information Bottleneck Co-clustering (aIBCC) approach, which automatically captures the relation between the numbers of clusters. The experimental results demonstrate the effectiveness and efficiency of our techniques.

Keywords: Co-clustering; Information Bottleneck

1 Introduction

Co-clustering [1] has emerged as an important approach for mining dyadic and relational data. Dyadic or relational observations are indexed by pairs of objects drawn from two index sets. For example, the index sets might be documents and words; the data might be incidence or counts of words in documents. Co-clustering allows documents and words to be grouped simultaneously and interdependently: documents are clustered based on the contained words, and words are grouped based on the documents in which they appear.

Some researchers have proposed a hard-partition version, Information Theoretic Co-clustering [2], others a soft-partition version [3, 4] of co-clustering. Usually, co-clustering algorithms are iterative, and an initialization of the clusters is required. The selection of a good initialization is a critical issue, since a random initialization often leads to sub-optimal solutions.

In this work, we propose a co-clustering approach based on the Information Bottleneck principle (Information Bottleneck Co-clustering, or IBCC). IBCC supports both soft-partition and hard-partition co-clustering. Furthermore, it uses an annealing-style strategy, called the continuation method [5], inspired by [7], which enables it to find near global optimal solutions. We also introduce an agglomerative Information Bottleneck Co-clustering (aIBCC) approach, which automatically captures the relation between the number of row- and column-clusters.

Information Bottleneck provides the foundation for a principled approach to co-clustering. As in [2], we view the data matrix as generated by a joint probability distribution between two random variables that are indexed by the rows and columns. Our approach finds row-clusters and column-clusters in an intertwined fashion so that the resulting clusters compress the data as much as possible, while preserving the relevant information. We make use of the formalism of Bayesian networks to specify the inter-dependencies between rows and columns of the data matrix, and thereby achieve the desired co-clustering.

While IBCC uses the same model as symmetric IB [8], our method also leverages the continuation method to achieve better solutions, and allows for both soft-partition and hard-partition clusterings. We also compare IBCC and Information Theoretic Co-clustering (ITCC) theoretically, and show that IBCC is an extension to ITCC in principle. Finally, we empirically analyze the relation between the number of row- and column-clusters.

2 Related Work

Information Bottleneck (IB) [6] is a powerful clustering approach, that leverages mutual information to evaluate how much information regarding the original data is kept by the clusters. It achieves a trade-off between

*Dept. of Computer Science, George Mason University

†Dept. of Systems Engineering and Operations Research, George Mason University

compressing the data and retaining as much information as possible about them. Multivariate IB [8] extends the original IB framework to handle the compression of multiple random variables. Although symmetric IB [8] uses the same model as IBCC, symmetric IB does not address co-clustering. Furthermore, IBCC uses the *continuation method* [5] to bypass local optima, and controls the softness of partitions via a parameter $\alpha \in (0, 1)$.

Elidan et al. [7] proposed the IB EM method to learn hidden variables in a graphical model. This method is based on Multivariate IB [8], an extension of the original IB approach. The IB EM algorithm also leverages the continuation method [5] to find near global optimal distributions for hidden variables.

Dhillon et al. [2] formulated co-clustering as an information theoretic problem. The data matrix is viewed as an empirical joint probability distribution of two discrete random variables. The optimal co-clustering maximizes the mutual information between clusters, subject to constraints on the numbers of row and column clusters.

An agglomerative hard clustering version of IB, called Double Clustering, was used in [17] to cluster documents after the clustering of words. The work in [18] extended the work in [17] to repetitively cluster documents and then words based on a greedy approach. In [2], it was empirically shown that information theoretic co-clustering is superior to Double Clustering [17], and comparable to its iterative extension [18]. In this paper, we show that our approach outperforms information theoretic co-clustering.

A generalized co-clustering framework in which any Bregman divergence can be used in the objective function has been introduced [16]. The statistics of the data to be preserved can be captured by properly defining conditional expectation-based constraints. An analysis of the co-clustering problem leads to the minimum Bregman information principle, and yields an algorithm that is guaranteed to achieve local optimality. Both methods [2, 16] perform hard-partition co-clustering.

Latent Dirichlet co-clustering [3] is a four-level hierarchical Bayesian model. Each document is modeled as a random mixture of topics, where each topic is a distribution over segments of the text. Each of these segments can be modeled as a mixture of word topics, where each topic is a distribution over words. The authors proposed an approximate inference technique based on the MCMC method, and a moment-matching algorithm for empirical Bayes parameter estimation.

Bayesian co-clustering [4] models a mixed membership in row and column clusters. This model maintains separate Dirichlet priors for rows and columns over the mixed membership, and assumes that each observation is generated by an exponential distribution corresponding to its row and column clusters. Shan et al. proposed a fast variational algorithm for inference of a Bayesian co-clustering model. This model, though, only supports soft-partitions. Both [3] and [4] are soft-partition co-clustering methods.

3 Preliminary

3.1 Information Bottleneck. The IB method [6] measures the relevance of a random variable X with respect to another random variable Y in terms of their mutual information [9]: $I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$. $I(X;Y)$ measures how many bits are needed on average to convey the information X has about Y, and vice versa.

Given the joint probability $P(X,Y)$, IB groups X into clusters T, with the aim of compressing X, while keeping as much information as possible about Y. Formally, IB minimizes the objective function: $\mathcal{L}[q(T|X)] = I(T;X) - \beta I(T;Y)$, where β is a positive Lagrange multiplier, and the minimization is w.r.t. all the normalized $q(T|X)$ distributions. The multiplier β implements a trade-off between compressing X and preserving the information about Y. β controls the softness of the clustering.

Since T is a compressed representation of X, its distribution should be completely determined given X alone; that is, $q(T|X,Y) = q(T|X)$, or $q(X,Y,T) = p(X,Y)q(T|X)$.

The distributions $q(T)$ and $q(Y|T)$ must satisfy the probabilistic consistency conditions [8]:

$$q(t) = \sum_x q(x,t) = \sum_x p(x)q(t|x) \quad (3.1)$$

$$q(y|t) = \frac{1}{q(t)} \sum_x q(x,y,t) = \frac{1}{q(t)} \sum_x p(x,y)q(t|x) \quad (3.2)$$

The stationary solution of the IB objective function

for a given β and number of clusters $|T|$ is:

$$q(t|x) = \frac{q(t)}{Z(x,\beta)} \exp(-\beta D_{KL}[p(y|x)\|q(y|t)]) \quad (3.3)$$

where $D_{KL}[p\|q] = E_p[\log p/q]$ is the Kullback-Leibler (KL) divergence [9] and $Z(x,\beta)$ is a normalization function. The three Eqs. (3.1), (3.2), and (3.3) must be satisfied self-consistently at all stationary solutions of the IB objective function $\mathcal{L}$. The distributions in Eqs. (3.1), (3.2), and (3.3) can be estimated in an iterative fashion. Specifically, the solutions $q(T|X)$, $q(T)$, $q(Y|T)$ correspond to a (local) stationary point of $\mathcal{L}$:

$$q^{(m)}(t) = \sum_x p(x) q^{(m)}(t|x) \quad (3.4)$$

$$q^{(m)}(y|t) = \frac{1}{q^{(m)}(t)} \sum_x p(x,y) q^{(m)}(t|x) \quad (3.5)$$

$$q^{(m+1)}(t|x) = \frac{q^{(m)}(t)}{Z^{(m+1)}(x,\beta)} \times \exp(-\beta D_{KL}[p(y|x)\|q^{(m)}(y|t)]) \quad (3.6)$$

3.2 Multivariate Information Bottleneck. Friedman et al. [8] further extended the original IB to a multivariate IB, which introduces the concept of multivariate information by using the semantics of Bayesian networks. Bayesian networks are used to highlight the relations implied by the data. In particular, one Bayesian network is used to define the information to be minimized between variables, and a second one is used to specify the information to be maximized.

A Bayesian network structure over a set of random variables $X_1, X_2, ..., X_n$ is a directed acyclic graph G in which vertices denote random variables. Parents of each variable X_i are represented as $Pa^G_{X_i}$. A distribution P consistent with G is such that $P(X_1, X_2, ..., X_n) = \prod_i P(X_i|Pa^G_{X_i})$.

Multi-information extends mutual information to the multivariate case. It represents the amount of information the variables $X_1, \ldots, X_n$ contain about each other, and is defined as follows: $I(X_1, \ldots, X_n) = D_{KL}[P(X_1, \ldots, X_n)\|P(X_1) \ldots P(X_n)] = E_P[\log \frac{P(X_1,\ldots,X_n)}{P(X_1)\ldots P(X_n)}]$. When P has additional known independencies, as in a Bayesian network G, the multi-information can be rewritten in terms of the dependencies among variables:

$$I(X_i, X_2, \ldots, X_n) = \sum_i I(X_i; Pa^G_{X_i}) \quad (3.7)$$

Following [8], we denote: $I^G = \sum_i I(X_i; Pa^G_{X_i})$.

Given a set of observed variables, $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$, multivariate IB considers a set of clusters $\mathbf{T} = \{T_1, T_2, \ldots, T_k\}$, which are different partitions of the observed variables. Specifically, multivariate IB constructs new cluster variables $\mathbf{T}$, where the relations between the observed variables and the cluster variables are expressed in a Bayesian network G^{in} over $\mathbf{X} \cup \mathbf{T}$. Once the relations are defined in G^{in}, the joint distribution over the observed variables and clusters is given by: $P(\mathbf{X}, \mathbf{T}) = P(\mathbf{X}) \prod_j P(T_j|Pa^{G^{in}}_j)$.

Similarly to the original IB formulation, $I^{G^{in}}$ embeds the information to be minimized. The minimization of $I^{G^{in}}$ attempts to make the variables in G^{in} as independent of each other as possible.

The information to be preserved by multivariate IB is specified by another Bayesian network, called G^{out}. G^{out} specifies which variables each T_j predicts. Each X_i or T_j is predicted by its parents in G^{out}. Thus, $I^{G^{out}}$ is a measure of how much information the variables $\mathbf{T}$ maintain about their target variables. In other words, $I^{G^{out}}$ is the quantity multivariate IB wants to maximize. Its generalized objective function is:

$$\mathcal{L}[P(T_1|Pa^{G^{in}}_{T_1}), ..., P(T_k|Pa^{G^{in}}_{T_k})] = I^{G^{in}} - \beta I^{G^{out}}$$

3.3 Agglomerative Multivariate Information Bottleneck. The agglomerative IB algorithm [14, 13, 19, 8] employs a greedy agglomerative clustering technique to find a hierarchical clustering tree in a bottom-up fashion. This algorithm starts with the most-fine grained solution, $T_j = Pa^{G^{in}}_{X_j}$, in which every $Pa^{G^{in}}_{X_j}$ is solely assigned to a unique singleton cluster, $t_j \in T_j$, and the assignment probabilities, $\{q(T_j|Pa^{G^{in}}_{X_j})\}^k_{j=1}$, are deterministic, that is their value is either 0 or 1.

Given the singleton initialization, the cardinality of one T_j is reduced by merging two of its values, t^l_j and t^r_j , into a single value $\bar{t}_j$. Formally, this is defined as $q(\bar{t}_j|Pa^{G^{in}}_{X_j}) = q(t^l_j|Pa^{G^{in}}_{X_j}) + q(t^r_j|Pa^{G^{in}}_{X_j})$.

The basic question in an agglomerative process is which pair to merge at each step. To this aim, it's more convenient to consider the problem of maximizing $\mathcal{L}_{max}(\{q(T_j|Pa^{G^{in}}_{X_j})\}^k_{j=1}) = I^{G^{out}} - \beta^{-1} I^{G^{in}}$.

Let T^{bef}_j and T^{aft}_j denote the random variables that correspond to T_j, before and after a merger in T_j,

respectively. Then, the merger cost is

$$\triangle\mathcal{L}_{max}(t_j^l, t_j^r) = \mathcal{L}_{max}^{bef} - \mathcal{L}_{max}^{aft} \tag{3.8}$$

where $\mathcal{L}_{max}^{bef}$ and $\mathcal{L}_{max}^{aft}$ are calculated based on T_j^{bef} and T_j^{aft}, respectively. A greedy procedure evaluates all the potential mergers, for every T_j, and applies the one that minimizes $\triangle\mathcal{L}_{max}(t_j^l, t_j^r)$. This is repeated until all the $\mathbf{T}$ variables degenerate into trivial clusters. The resulting set of hierarchies describes a range of solutions at different resolutions.

A direct calculation of all the potential merger costs using Eq. (3.8) is typically unfeasible. However, as in [6], one may calculate $\triangle\mathcal{L}_{max}(t_j^l, t_j^r)$ while examining only the distributions that involve t_j^l and t_j^r directly [8].

4 Information Bottleneck Co-clustering

We leverage the framework provided by multivariate IB to perform co-clustering. To introduce our Information Bottleneck Co-clustering (IBCC) algorithm, we assume we are dealing with text data, which is also the focus of our experiments (but IBCC can be applied to other domains, not only text, such as movie-review data). Thus, a document-word matrix is constructed, where each row is a document and each column is a word. The element of position (n, m) represents the joint empirical probability $p(w_m, d_n)$, where w_m is the m-th word and d_n is the n-th document.

4.1 IBCC for Fixed α. Let W and D be two discrete random variables that take values over the words and documents, respectively. Given a word-document joint distribution $P(W, D)$, IBCC is to group both words and documents into clusters based on their relations. Figure 1 gives the Bayesian networks G^{in} and G^{out} that capture the dependency relations we desire. $\tilde{W}$ and $\tilde{D}$ are two discrete random variables that take values over the word clusters and document clusters, respectively. G^{in} specifies that $\tilde{W}$ is to group W, $\tilde{D}$ is to group D, and there is a dependency between W and D. G^{out} specifies that $\tilde{W}$ predicts D, $\tilde{D}$ predicts W, and there is a dependency between $\tilde{W}$ and $\tilde{D}$[1]. Thus, G^{in} and G^{out} highlight the dependencies between $\tilde{W}$, $\tilde{D}$, W, and D to achieve the desired co-clustering.

[1]The arrow direction between W and D in G^{in}, and between $\tilde{W}$ and $\tilde{D}$ in G^{out}, is irrelevant, since W and D are symmetric (in [8], this property is called symmetric IB). This phenomenon applies also to the solution of IBCC.

According to Eq. (3.7), the multi-information of G^{in} is $I^{G^{in}} = I(W;\tilde{W}) + I(D;\tilde{D}) + I(W;D)$, and the multi-information of G^{out} is $I^{G^{out}} = I(W;\tilde{D}) + I(D;\tilde{W}) + I(\tilde{W};\tilde{D})$. Following the IB principle, we can obtain the objective function of IBCC:

$$\begin{aligned}\mathcal{L}[q(\tilde{w}|w), q(\tilde{d}|d)] = \\ (I(W;\tilde{W}) + I(D;\tilde{D}) + I(W;D)) - \\ \beta(I(W;\tilde{D}) + I(D;\tilde{W}) + I(\tilde{W};\tilde{D}))\end{aligned} \tag{4.9}$$

$I(W;D)$ is a constant and can be ignored. β controls the softness of the partitions, and its range is unbounded: $(0, +\infty)$. In practice, it is convenient to have a bounded parameter. Thus, we introduce $\alpha \in (0, 1)$:

$$\begin{aligned}\mathcal{L}[q(\tilde{w}|w), q(\tilde{d}|d)] = \\ (1-\alpha)(I(W;\tilde{W}) + I(D;\tilde{D})) - \\ \alpha(I(W;\tilde{D}) + I(D;\tilde{W}) + I(\tilde{W};\tilde{D}))\end{aligned}$$

The stationary solution of $\min \mathcal{L}[q(\tilde{w}|w), q(\tilde{d}|d)]$, for a fixed α and numbers of clusters $|\tilde{W}|$ and $|\tilde{D}|$ is [8]:

$$\begin{aligned}q(\tilde{w}_j|w_i) = \\ \frac{q(\tilde{w}_j)}{Z(w_i,\alpha)} \exp\{-\frac{\alpha}{1-\alpha}(D_{KL}[q(D|w_i)\|q(D|\tilde{w}_j)] \\ +D_{KL}[q(\tilde{D}|w_i)\|q(\tilde{D}|\tilde{w}_j)])\}\end{aligned} \tag{4.10}$$

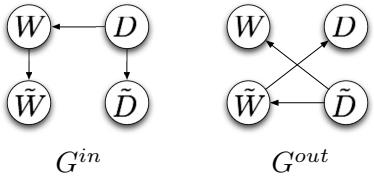
$$\begin{aligned}q(\tilde{d}_j|d_i) = \\ \frac{q(\tilde{d}_j)}{Z(d_i,\alpha)} \exp\{-\frac{\alpha}{1-\alpha}(D_{KL}[q(W|d_i)\|q(W|\tilde{d}_j)] \\ +D_{KL}[q(\tilde{W}|d_i)\|q(\tilde{W}|\tilde{d}_j)])\}\end{aligned} \tag{4.11}$$

where the $Z(\cdot,\cdot)$s are probability normalization factors:

$$\begin{aligned}Z(w_i,\alpha) = \\ \sum_{m=1}^{K} q(\tilde{w}_m) \exp\{-\frac{\alpha}{1-\alpha}(D_{KL}[q(D|w_i)\|q(D|\tilde{w}_m)] \\ +D_{KL}[q(\tilde{D}|w_i)\|q(\tilde{D}|\tilde{w}_m)])\}\end{aligned}$$

$$\begin{aligned}Z(d_i,\alpha) = \\ \sum_{m=1}^{K} q(\tilde{d}_m) \exp\{-\frac{\alpha}{1-\alpha}(D_{KL}[q(W|d_i)\|q(W|\tilde{d}_m)] \\ +D_{KL}[q(\tilde{W}|d_i)\|q(\tilde{W}|\tilde{d}_m)])\}\end{aligned}$$

$q(\tilde{w}_j|w_i)$ ($q(\tilde{d}_j|d_i)$, respectively) defines the probability that a word w_i (document d_i) belongs to a word



G^{in} G^{out}

Figure 1: G^{in} and G^{out} for IBCC

Algorithm 1 IBCC for Fixed α

Input: Document-word joint probability matrix, word and document clusters' sizes $|\tilde{W}|$ and $|\tilde{D}|$, initial clusters' distributions $Q^{(0)}(\tilde{W}|W)$ and $Q^{(0)}(\tilde{D}|D)$, and α
Output: Co-clustering results $Q^{(m)}(\tilde{W}|W)$ and $Q^{(m)}(\tilde{D}|D)$
begin
m=0
repeat

- Compute marginal and conditional probabilities in Eqs. (3.4)-(3.6) using $Q^{(m)}(\tilde{W}|W)$ and $Q^{(m)}(\tilde{D}|D)$
- Calculate $Q^{(m+1)}(\tilde{W}|W)$ and $Q^{(m+1)}(\tilde{D}|D)$ using Eqs. (4.10) and (4.11)
- Set $m = m + 1$

until Convergence of $Q(\tilde{W}|W)$ and $Q(\tilde{D}|D)$.
end

cluster $\tilde{w}_j$ (document cluster $\tilde{d}_j$). Thus, the distribution $Q(\tilde{W}|W)$ ($Q(\tilde{D}|D)$) gives a partition[2] of words W (documents D).

Again, the solution of IBCC can be calculated iteratively. After computing $q^{(m)}(\tilde{w}_j|w_i)$ and $q^{(m)}(\tilde{d}_j|d_i)$, we can compute the other marginal and conditional probabilities in turn. We can then compute the updates $q^{(m+1)}(\tilde{w}_j|w_i)$ and $q^{(m+1)}(\tilde{d}_j|d_i)$ using Eqs. (4.10) and (4.11).

Algorithm 1 summarizes the IBCC algorithm for a fixed α. Convergence is achieved when the Jensen-Shannon (JS) divergences between $Q^{(m+1)}(\tilde{W}|W)$ and $Q^{(m)}(\tilde{W}|W)$, and between $Q^{(m+1)}(\tilde{D}|D)$ and $Q^{(m)}(\tilde{D}|D)$, are smaller than a threshold [8].

From Eqs. (4.10) and (4.11), we can see that IBCC does leverage the relations between words and documents to perform clustering. $D_{KL}[q(D|w_i)\|q(D|\tilde{w}_j)]$ evaluates the distance between the i-th word w_i and the j-th word cluster $\tilde{w}_j$ based on all documents; $D_{KL}[q(\tilde{D}|w_i)\|q(\tilde{D}|\tilde{w}_j)]$ evaluates the distance between w_i and $\tilde{w}_j$ based on all document clusters. The smaller the sum of the two KL divergences, the larger the probability of word cluster $\tilde{w}_j$ given word w_i. Thus, the probability $q(\tilde{w}_j|w_i)$ of the j-th word cluster $\tilde{w}_j$ given the i-th word w_i depends on all the documents and document clusters. Similarly for the probability $q(\tilde{d}_j|d_i)$ of the j-th document cluster $\tilde{d}_j$ given the i-th document d_i.

In Figure 1, G^{out} shows that document clusters can predict word clusters, that is, documents grouped in the same document cluster contain words from the same word cluster. This means that documents of similar topics may have similar words. Similarly, word clusters can also predict document clusters: similar words often appear in documents of similar topics. As a consequence, as mentioned earlier, the arrow direction between $\tilde{W}$ and $\tilde{D}$ in G^{out} is irrelevant. The same holds for the arrow between W and D in G^{in}. From Eqs. (4.10) and (4.11), it is evident that W and D (as well as $\tilde{W}$ and $\tilde{D}$) are symmetric.

4.2 Bypassing Local Optima. Eqs. (4.10) and (4.11) provide a solution of IBCC for a fixed value of α. The quality of the solution depends on the initialization. Given a random initialization, typically the resulting clusters correspond to local optima.

We adopt a more refined approach, called *continuation method* [5], to conduct an annealing strategy that allows to bypass local minima. To perform continuation, we view the optimization problem in the space of α. In this space, we want to find a smooth path that starts from the trivial solution at $\alpha = 0$, and goes through stationary point solutions for different α values. Continuation theory guarantees that such path, free of discontinuities, exists. For a very small value of α, the initialization of clusters to the uniform distribution is very close to the optimal solution of IBCC. α is then gradually increased, and the solution of IBCC for the previous α value is used as initialization for the current α, and the new stationary point is calculated. The process is iterated until α is close to 1. At the beginning, α is close to 0, so the probabilities of word w/document d belonging to all word/document clusters are almost equal. As α slowly increases, the peak(s) of $q(\tilde{w}_j|w_i)$ and $q(\tilde{d}_j|d_i)$ gradually show(s) up, and become trivial singleton clusters when α approaches 1. Thus, this procedure needs to stop at a proper value of $\alpha \in (0, 1)$.

A problem arises: how to choose the proper step length for changing α. There are various standard approaches, such as normalizing the direction vector

[2]The partition may be soft or hard, according to the value of α.

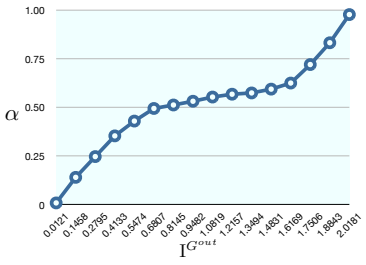


Figure 2: Step Length

Algorithm 2 Information Bottleneck Co-clustering

Input: Document-word joint probability matrix, word and document clusters' sizes $|\tilde{W}|$ and $|\tilde{D}|$, and initial small value of α
Output: Co-clustering results $Q(\tilde{W}|W)$ and $Q(\tilde{D}|D)$ for $\alpha \in (0, 1)$
begin
Initialize $Q(\tilde{W}|W)$ and $Q(\tilde{D}|D)$ as uniform distributions
repeat

- Use current $Q(\tilde{W}|W)$ and $Q(\tilde{D}|D)$ as initialization to calculate the solution of IBCC $Q'(\tilde{W}|W)$ and $Q'(\tilde{D}|D)$ at α using Algorithm 1
- Calculate the step length $\triangle\alpha$ for the current α using Eq. (4.12)
- Update $\alpha = \alpha + \triangle\alpha$
- Set $Q(\tilde{W}|W) = Q'(\tilde{W}|W)$ and $Q(\tilde{D}|D) = Q'(\tilde{D}|D)$

until $\alpha \geq 1$
end

to a predetermined size. Following [7], we adopt a more natural measure of the step length of α as follows. Recall that $I^{G^{out}} = I(W;\tilde{D}) + I(D;\tilde{W}) + I(\tilde{W};\tilde{D})$, by increasing α, IBCC puts more emphasis on $I^{G^{out}}$. As such, the word and document clusters $\tilde{W}$ and $\tilde{D}$ capture more information, and $I^{G^{out}}$ becomes larger. Figure 2 gives $I^{G^{out}}$ as a function of α. It shows that the multi-information $I^{G^{out}}$ is an increasing function of α, and its rate of change varies at different α values. Thus, we calibrate the change of α using $I^{G^{out}}$. It is clear that $I(W;\tilde{D}) \leq I(W;D)$, $I(D;\tilde{W}) \leq I(W;D)$ and $I(\tilde{W};\tilde{D}) \leq I(W;D)$, and therefore, $0 \leq I^{G^{out}} \leq 3I(W;D)$. We fix the change of $I^{G^{out}}$, denoted as $\triangle I^{G^{out}}$, and then calculate $\frac{\partial I^{G^{out}}}{\partial\alpha}$ for the current α. The step length of α is:

$$\triangle\alpha = \frac{\partial I^{G^{out}}}{\partial\alpha}\triangle I^{G^{out}} \tag{4.12}$$

where $\frac{\partial I^{G^{out}}}{\partial\alpha} = \frac{\partial I(W,\tilde{D})}{\partial\alpha} + \frac{\partial I(D,\tilde{W})}{\partial\alpha} + \frac{\partial I(\tilde{W},\tilde{D})}{\partial\alpha}$. Algorithm 2 summarizes the IBCC method.

5 Agglomerative IBCC

Here we leverage the framework provided by agglomerative multivariate IB to perform co-clustering. As introduced in Section 3.3, it is more convenient to formulate agglomerative IB as a maximization problem. Thus, the objective function of aIBCC can be rewritten as:

$$\begin{aligned}\mathcal{L}_{max}[q(\tilde{w}|w), q(\tilde{d}|d)] = \\ (I(W;\tilde{D}) + I(D;\tilde{W}) + I(\tilde{W};\tilde{D})) - \\ \beta^{-1}(I(W;\tilde{W}) + I(D;\tilde{D}))\end{aligned} \tag{5.13}$$

The merging cost of two word-clusters $\tilde{w}_l$ and $\tilde{w}_r$, or two document-clusters $\tilde{d}_l$ and $\tilde{d}_r$, is [8]:

$$\triangle\mathcal{L}_{max}(\tilde{w}_l, \tilde{w}_r) = q(\tilde{w}_{l+r}) \cdot d(\tilde{w}_l, \tilde{w}_r) \tag{5.14}$$

$$\triangle\mathcal{L}_{max}(\tilde{d}_l, \tilde{d}_r) = q(\tilde{d}_{l+r}) \cdot d(\tilde{d}_l, \tilde{d}_r) \tag{5.15}$$

where

$$\begin{aligned}d(\tilde{w}_l, \tilde{w}_r) = JS_\Pi[q(D|\tilde{w}_l), q(D|\tilde{w}_r)] + \\ JS_\Pi[q(\tilde{D}|\tilde{w}_l), q(\tilde{D}|\tilde{w}_r)] - \\ \beta^{-1} JS_\Pi[q(W|\tilde{w}_l), q(W|\tilde{w}_r)]\end{aligned} \tag{5.16}$$

$$\begin{aligned}d(\tilde{d}_l, \tilde{d}_r) = JS_\Pi[q(W|\tilde{d}_l), q(W|\tilde{d}_r)] + \\ JS_\Pi[q(\tilde{W}|\tilde{d}_l), q(\tilde{W}|\tilde{d}_r)] - \\ \beta^{-1} JS_\Pi[q(D|\tilde{d}_l), q(D|\tilde{d}_r)]\end{aligned} \tag{5.17}$$

where JS_Π is the Jensen-Shannon (JS) divergence $JS_\Pi[q_1, q_2] = \pi_1 D_{KL}(q_1\|\bar{q}) + \pi_2 D_{KL}(q_2\|\bar{q})$ and $\Pi = \{\pi_1, \pi_2\}$, $\pi_1 + \pi_2 = 1$, $\bar{q} = \pi_1 q_1 + \pi_2 q_2$.

In our case, for word-clusters, $\pi_1 = q(\tilde{w}_l)/q(\tilde{w}_{l+r})$ and $\pi_2 = q(\tilde{w}_r)/q(\tilde{w}_{l+r})$; for document-clusters, $\pi_1 = q(\tilde{d}_l)/q(\tilde{d}_{l+r})$ and $\pi_2 = q(\tilde{d}_r)/q(\tilde{d}_{l+r})$. These are the relative weights of each of the clusters that participate in the merged cluster.

Since, in this work, we address hard-partition, the conditional probabilities, $q(w|\tilde{w}_l)$, $q(w|\tilde{w}_r)$, $q(d|\tilde{d}_l)$ and $q(d|\tilde{d}_r)$ are deterministic, i.e., their value is either 1 or 0. So $JS_\Pi[q(W|\tilde{w}_l), q(W|\tilde{w}_r)] = H(\Pi) = q(\tilde{w}_l)\log q(\tilde{w}_l) + q(\tilde{w}_r)\log q(\tilde{w}_r)$, where $H(\Pi)$ is Shannon's entropy.

The merging cost of two word- or document-clusters is the product of the "weight" of the merger components, $q(\tilde{w}_{l+r})$ or $q(\tilde{d}_{l+r})$, with their "distance" $d(\tilde{w}_l, \tilde{w}_r)$ or $d(\tilde{d}_l, \tilde{d}_r)$. Due to the JS properties, This "distance" is symmetric, but it is not a metric. It is

Algorithm 3 Agglomerative Information Bottleneck Co-clustering (aIBCC)

Input: Document-word joint probability matrix, initial word-clusters $\tilde{W}_i$ and document clusters $\tilde{D}_i$, desired number of document-clusters $|\tilde{D}_o|$, and γ.
Output: Partition of W into $|\tilde{W}_o|$ clusters; partition of D into $|\tilde{D}_o|$ clusters (corresponding to the clustering structure at the proper level of the resulting tree).
begin
Calculate the merging cost between all pairs of word-clusters in $\tilde{W}_o$ and between all pairs of document-clusters in $\tilde{D}_o$ using Eqs. (5.14) and (5.15) respectively.
repeat
- Find the minimal merging cost of word-clusters $\triangle\mathcal{L}_{max}(\tilde{w}_l, \tilde{w}_r)$ and of document-clusters $\triangle\mathcal{L}_{max}(\tilde{d}_l, \tilde{d}_r)$
- **if** $\triangle\mathcal{L}_{max}(\tilde{w}_l, \tilde{w}_r) > \gamma \cdot \triangle\mathcal{L}_{max}(\tilde{d}_l, \tilde{d}_r)$, **then** merge two document-clusters $\tilde{d}_l$ and $\tilde{d}_r$ into one cluster $\tilde{d}_{l+r}$, update the merging cost between $\tilde{d}_{l+r}$ and all other document-clusters
- **else** merge two word-clusters $\tilde{w}_l$ and $\tilde{w}_r$ into one cluster $\tilde{w}_{l+r}$, update the merging cost between $\tilde{w}_{l+r}$ and all other word-clusters

until $|\tilde{D}| = 1$ or $|\tilde{W}| = 1$
end

small for pairs of words that co-occur frequently in similar documents, or for pairs of documents about similar topics, which means that the documents may contain similar words.

The agglomerative IBCC method is shown in Algorithm 3. At each step, aIBCC merges the two clusters with minimal merging cost. The parameter γ is a trade-off parameter used to weigh the costs of word-clusters and document-clusters. The initial clusters of aIBCC may be singleton clusters, or small-sized clusters. In practice, for text data, the initial number of word-clusters is typically larger than the number of document-clusters. As such, the minimal cost of merging two word-clusters tends to be smaller than that of merging two document-clusters. As a result, aIBCC merges word-clusters more frequently than document-clusters, and, typically, $|\tilde{D}|$ approaches one before $|\tilde{W}|$ does, for $\gamma = 1$. This mechanism allows the number of word-clusters to adapt to the number of document-clusters, and vice versa, regardless of the number of initial clusters. Figure 3 shows this behavior for the 20 Newsgroups data "rec vs talk".

The complexity of Eqs. (5.14) and (5.15) is $O(|D|+|\tilde{D}|) = O(|D|)$ and $O(|W|+|\tilde{W}|) = O(|W|)$, respectively, since $|D| >> |\tilde{D}|$ and $|W| >> |\tilde{W}|$. Suppose the numbers of input word-clusters and document-clusters are $|\tilde{W}_o| = l$ and $|\tilde{D}_o| = m$, respectively. The computation of the cost of merging all pairs of word-clusters and document-clusters has a complexity $O(l^2|D|) + O(m^2|W|)$. When two word-clusters (two document-clusters) are merged, the update of the merging cost between the newly merged cluster and all the other clusters needs at most $O(l|D|)$ $(O(m|W|))$ operations. Thus, the overall complexity of aIBCC is $O(l^2|D| + m^2|W|)$.

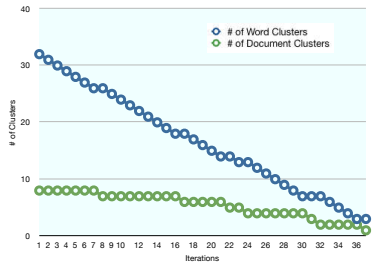


Figure 3: # Word Clusters vs. # Document Clusters

6 Analysis of IBCC

In this section, we analyze the differences between the IBCC algorithm proposed here, and the Information Theoretic Co-clustering (ITCC) approach [2].

The objective function minimized by ITCC is:

$$I(W;D) - I(\tilde{W};\tilde{D}) \tag{6.18}$$

By comparing Eqs. (6.18) and (4.9), we can observe that IBCC explicitly takes into account the two mutual information measures $I(W;\tilde{D})$ and $I(D;\tilde{W})$, while ITCC accounts only for $I(\tilde{W};\tilde{D})$. As a result IBCC may force document-clusters to incorporate information about words, and word-clusters to incorporate information about documents, thus capturing the "essence" of co-clustering.

According to [8], the stationary point solution of the multivariate IB objective function, Eq. (4.9), is:

$$q(\tilde{w}_j|w_i) = \frac{q(\tilde{w}_j)}{Z(w_i,\beta)} \exp\{-\beta(D_{KL}[q(D|w_i)\|q(D|\tilde{w}_j)] + D_{KL}[q(\tilde{D}|w_i)\|q(\tilde{D}|\tilde{w}_j)])\} \tag{6.19}$$

$$q(\tilde{d}_j|d_i) = \frac{q(\tilde{d}_j)}{Z(d_i,\beta)} \exp\{-\beta(D_{KL}[q(W|d_i)\|q(W|\tilde{d}_j)] + D_{KL}[q(\tilde{W}|d_i)\|q(\tilde{W}|\tilde{d}_j)])\} \tag{6.20}$$

Theoretically, IBCC provides clusters according to Eqs. (6.19) and (6.20), which give the probability of a

word/document cluster given a word/document. Since we only consider hard-partitions here, probabilities are either 1 or 0. The cost of merging two clusters, expressed in Eqs. (5.14) and (5.15), originates from a change in the partition induced by merging clusters. Thus, IBCC does not compute Eqs. (6.19) and (6.20) explicitly, but calculates the merging costs through Eqs. (5.14) and (5.15), and generates the final partitions following the agglomerative strategy.

The solution of ITCC is:

$$I(W;D) - I(\tilde{W};\tilde{D}) = D_{KL}[p(W,D)\|q(W,D)] \quad (6.21)$$

where $q(w,d) = p(w,d)p(w|\tilde{w})p(d|\tilde{d})$ for $w \in \tilde{w}$ and $d \in \tilde{d}$. Since ITCC gives a hard-partition, it can be rewritten:

$$\begin{aligned} I(W;D) - I(\tilde{W};\tilde{D}) = \\ D_{KL}[p(W,D,\tilde{W},\tilde{D})\|q(W,D,\tilde{W},\tilde{D})] = \\ \sum_{\tilde{w}} \sum_{w \in \tilde{w}} p(w) D_{KL}[p(D|w)\|q(D|\tilde{w})] = \quad (6.22) \\ \sum_{\tilde{d}} \sum_{d \in \tilde{d}} p(d) D_{KL}[p(W|d)\|q(W|\tilde{d})] \quad (6.23) \end{aligned}$$

where $q(w|\tilde{d}) = p(w|\tilde{w})p(\tilde{w}|\tilde{d})$ and $q(d|\tilde{w}) = p(d|\tilde{d})p(\tilde{d}|\tilde{w})$. For each w or d, minimizing $D_{KL}[p(D|w)\|q(D|\tilde{w})]$ or $D_{KL}[p(W|d)\|q(W|\tilde{d})]$ allows to minimize the objective function (6.21). Thus, ITCC assigns w or d to $\tilde{w}$ or $\tilde{d}$, respectively, with minimal "distance", that is:

$$C(w) = \arg\min_{\tilde{w}} D_{KL}[p(D|w)\|q(D|\tilde{w})] \quad (6.24)$$

$$C(d) = \arg\min_{\tilde{d}} D_{KL}[p(W|d)\|q(W|\tilde{d})] \quad (6.25)$$

where $C(w)$ or $C(d)$ represents the assigned word/document cluster for w or d. Note that $D_{KL}[p(D|w)\|q(D|\tilde{w})]$ in Eq. (6.24) is equal to $D_{KL}[q(D|w_i)\|q(D|\tilde{w}_j)]$ in Eq. (4.10), and the same holds for $D_{KL}[p(W|d)\|q(W|\tilde{d})]$ in Eq. (6.25) and $D_{KL}[q(W|d_i)\|q(W|\tilde{d}_j)]$ in Eq. (4.11).

Comparing Eqs. (4.10) and (4.11), and Eqs. (6.24) and (6.25), we can see that when IBCC assigns a word w to a word-cluster $\tilde{w}$, it measures the KL-divergence between w and $\tilde{w}$ not only in terms of the documents D ($D_{KL}[q(D|w)\|q(D|\tilde{w})]$), but also in terms of the document-clusters $\tilde{D}$ ($D_{KL}[q(\tilde{D}|w)\|q(\tilde{D}|\tilde{w})]$). On the other hand, ITCC only considers the KL-divergence based on D. A similar argument holds for the KL-divergence between d and $\tilde{d}$. In this prospective, the IBCC algorithm can be seen as an extension of the ITCC approach.

Table 1: Data Sets

	Data Sets	Combinations
20 Newsgroups	rec vs talk vs sci	rec.sport.baseball rec.sport.hockey talk.politics.mideast talk.religion.misc sci.crypt sci.electronics
	sci vs talk vs comp	sci.space sci.med talk.politics.misc talk.politics.guns comp.os.ms-windows.misc comp.sys.ibm.pc.hardware
	comp vs sci	comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x sci.med sci.space
	rec vs talk	rec.sport.baseball rec.sport.hockey talk.politics.mideast talk.religion.misc
	rec vs sci	rec.motorcycles rec.sport.hockey sci.crypt sci.electronics
	sci vs talk	sci.crypt sci.space talk.politics.guns talk.politics.mideast
SRAA	auto vs aviation	real-auto real-aviation
	real vs simulated	real-auto sim-auto

7 Experimental Results

7.1 Data Sets. We evaluated our algorithm using two real data sets: 20 Newsgroups [10] and SRAA [11]. We generated six data sets from 20 Newsgroups. SRAA is a Simulated/Real/Aviation/Auto UseNet data set for classication. It contains 73,218 articles from four discussion groups: simulated auto racing, simulated aviation, real autos, and real aviation. Table 1 provides the details for both data sets.

7.2 Implementation Details. Pre-processing was performed on the raw data. All letters were converted to a lower case, words were stemmed using the Porter algorithm, and stop words were removed. We used a simple feature selection method, Document Frequency

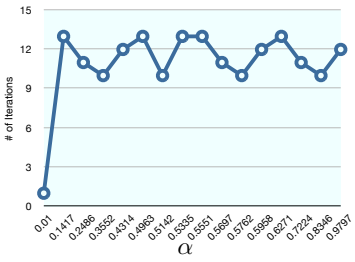


Figure 4: IBCC: # of iterations vs. α

(DF) thresholding, to cut down the number of features. The DF threshold was set to 3. TF-IDF was used for feature weighting.

A word may not appear in every document in a given corpus. As such, the joint probability of a word and a document may become zero. This can lead to instabilities when calculating the KL divergence in Eqs. (4.10) and (4.11) and the JS divergence in Eqs. (5.16) and (5.17). We adopt Laplace smoothing [12] to cope with this problem.

Although the continuation method may help IBCC avoid suboptimal solutions, the approximation, as well as inherent numerical instability, can lead to a suboptimal path [7]. To cope with this, we adopt a commonly used heuristic in deterministic annealing: at each value of α, we slightly perturb the initial solution before starting the IBCC iterations.

As described in Section 4.2, $0 \leq I^{G^{out}} \leq 3I(W;D)$. We set $\triangle I^{G^{out}} = 3I(W;D)/1000$. In practice, the largest value of $I^{G^{out}}$ is far less than $3I(W;D)$, since $|\tilde{W}| << |W|$ and $|\tilde{D}| << |D|$. Thus, for a given value of α, IBCC usually converges after a small number of iterations. Figure 4 shows this behavior on the data set "rec vs talk".

Although aIBCC may use initial singleton clusters, this scenario is infeasible for large data sets: for the 20 Newsgroups $|W| = 12,000$, and for SRAA $|W| = 15,000$. Thus, the computation of all pairwise merging costs for singleton clusters becomes too expensive. We therefore group words and documents independently into clusters using Cluto [15]. The resulting clusters are provided in input as initialization to aIBCC.

7.3 Performance. IBCC computes a stationary solution for each α value along the path between 0 and 1. Ultimately, a single solution needs to be chosen, and the proper choice depends on the application. In this work, we considered the solution corresponding to the largest value of α, which gives a hard-partition.

We evaluated IBCC and aIBCC w.r.t. document clustering precision computed according to class labels. IBCC provides a soft-partition: $Q(\tilde{D}|D)$ gives the probability of each document belonging to each cluster. To compute precision, a document is assigned to the cluster that provides the largest probability. For the 20 Newsgroups, we set $|\tilde{W}| = 32$ and $|\tilde{D}| = 4$, with the exception of $|\tilde{D}| = 5$ for "comp vs sci"; for SRAA, $|\tilde{D}| = 2$ and $|\tilde{W}| = 64$.

We also implemented the Information Theoretic Co-clustering (ITCC) [2] algorithm, and report the precision it achieved at the 10^{th} iteration (in all cases, 10 iterations were enough to achieve convergence). For a fair comparison, we used Cluto to initialize the word- and document-clusters for ITCC as well. The initial number of document-clusters is set to the actual number of classes in the data. To set the numbers of initial word-clusters for ITCC, we consider the number of word-clusters computed by the solution of aIBCC. In this way, the solutions provided by the two algorithms contain the same numbers of word- and document-clusters. Table 4 provides the initial numbers of word-clusters ($|\tilde{W}_i|$) and document-clusters ($|\tilde{D}_i|$) given in input to aIBCC for each data set. $|\tilde{W}_o|$ and $|\tilde{D}_o|$ represent the numbers of word-clusters and document-clusters of the solution computed by aIBCC. Thus, $|\tilde{W}_o|$ and $|\tilde{D}_o|$ are also the numbers of clusters given in input to ITCC.

Table 2 shows the precision obtained for each data set. (We didn't run IBCC on the two multi-class sets because the execution time would have been too long.) "IBCC w/ CM" means that the continuation method was used to calibrate the step length of α; "IBCC w/o CM" means that IBCC was performed from a random initialization at the corresponding largest α, without the use of the continuation method. High precision values were obtained in each case, confirming the feasibility of IBCC and aIBCC. The precision achieved by aIBCC is superior to that achieved by ITCC for each data set. The use of the continuation method in IBCC improves the precision in each case. Figure 6 shows the precision values as a function of α for the data set "rec vs talk"; it demonstrates that the continuation method does help IBCC finding better stationary solutions, with respect to the random initialization strategy, for each α values.

Figure 5 shows the relation between precision and

Table 2: Clustering Precision

Data Sets	Precision				
	IBCC w/ CM	IBCC w/o CM	aIBCC	ITCC	CLUTO
rec vs talk vs sci	-	-	0.806	0.759	0.689
sci vs talk vs comp	-	-	0.819	0.767	0.707
comp vs sci	0.785	0.756	0.802	0.759	0.695
rec vs talk	0.876	0.825	0.904	0.883	0.815
rec vs sci	0.848	0.809	0.876	0.797	0.728
sci vs talk	0.907	0.853	0.869	0.806	0.763
auto vs aviation	0.871	0.753	0.897	0.852	0.828
real vs simulated	0.833	0.706	0.841	0.808	0.764

Table 3: Running Time

Data Sets	Time (min)	
	aIBCC	ITCC
rec vs talk vs sci	80.36	829.97
sci vs talk vs comp	78.49	834.58
comp vs sci	25.37	304.76
rec vs talk	23.12	281.39
rec vs sci	22.08	257.53
sci vs talk	22.81	241.49
auto vs aviation	105.28	1099.27
real vs simulated	109.73	1130.65

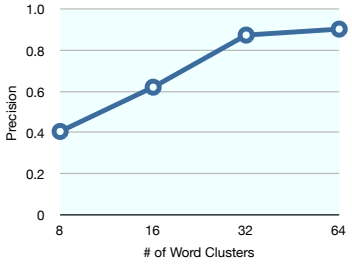


Figure 5: Precision vs. # word clusters

Table 4: aIBCC: # Word- and Document-clusters

Data Sets	$\lvert\tilde{W}_i\rvert$	$\lvert\tilde{D}_i\rvert$	$\lvert\tilde{W}_o\rvert$	$\lvert\tilde{D}_o\rvert$
rec vs talk vs sci	48	16	16	6
sci vs talk vs comp	48	16	16	6
comp vs sci	32	8	15	5
rec vs talk	32	8	13	4
rec vs sci	32	8	12	4
sci vs talk	32	8	11	4
auto vs aviation	64	8	14	2
real vs simulated	64	8	13	2

Table 5: aIBCC: Precision for different initializations

$\lvert\tilde{W}_i\rvert$	$\lvert\tilde{D}_i\rvert$	Precision
64	8	0.904
32	8	0.904
16	8	0.904
8	8	0.821
16	16	0.911
32	16	0.911
32	32	0.916
64	32	0.916

the # of word clusters for "rec vs talk". Higher precision is obtained for a larger number of word clusters.

As discussed in Section 4.2, we choose $I^{G^{out}}$ to scale the step length of α. Figure 2 shows the relation between $I^{G^{out}}$ and α. When α is close to 0 or 1, large changes of α correspond to small changes of $I^{G^{out}}$. For mid-range values of α, small changes of α correspond to large changes of $I^{G^{out}}$. Thus, by fixing the variation of $I^{G^{out}}$, we can properly calibrate the change in α.

Table 3 shows the running times of aIBCC and ITCC. The reported times for aIBCC include the execution of Cluto. For aIBCC, γ was set to 1.0. aIBCC is more than 10 times faster than ITCC. Although aIBCC is quadratic and ITCC is linear w.r.t. the number of data, aIBCC only needs to deal with the clusters initialized by Cluto, while ITCC still needs to deal with each document. In all cases, aIBCC was able to achieve good precision rates, while effectively reducing the dimensionality in an adaptive fashion.

We also evaluated the performance of aIBCC with respect to different initializations. Table 5 shows the precision achieved by aIBCC on "rec vs talk", for $\gamma = 1.0$ and different combinations of values for $|\tilde{W}_i|$ and $|\tilde{D}_i|$. Overall, aIBCC is quite stable, with an increase in precision for a larger number of initial clusters. In particular, we observe a degradation in precision when the initial number of word-clusters is set to a value ($|\tilde{W}_i| = 8$) less than the "optimal" one achieved by aIBCC ($|\tilde{W}_o| = 13$, see Table 4). Figure 7 shows the precision as a function of different γ values for the "rec vs talk" data ($|\tilde{W}_i| = 32$ and $|\tilde{D}_i| = 8$). γ can control the frequency according to which word-clusters and document-clusters are merged. If γ is small, aIBCC merges document-clusters more frequently; when γ is large, aIBCC merges word-clusters more frequently. $\gamma = 1.0$ achieves the best precision in this case.

Finally, we report the relation between the merging cost and the # of clusters. Intuitively, smaller numbers of clusters lead to larger merging costs. Figure 8 shows the smallest cost of merging two word-clusters with respect to the # of word-clusters at different iterations. Figure 9 shows the smallest cost of merging two document-clusters with respect to the # of document clusters at different iterations. Both figures are based on the data "rec vs talk" with $\gamma = 1.0$, and initial numbers of clusters $|\tilde{W}_i| = 32$ and $|\tilde{D}_i| = 8$. Figure 9

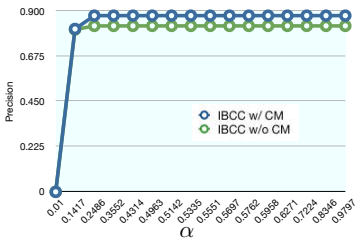


Figure 6: IBCC: Precision vs. α

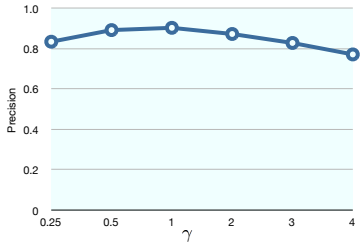


Figure 7: aIBCC: Precision vs. γ

shows that merging two document-clusters has a high cost. This is expected, since this data set is constructed from two top categories which differ in content.

8 Conclusions

We introduced two co-clustering algorithms, and experimentally demonstrated their effectiveness. The analysis of merging costs at different iterations of aIBCC shows potential for the design of a strategy to automatically set the number of document-clusters in the data.

Acknowledgments

This work is in part supported by NSF CAREER Award IIS-0447814.

References

[1] J. A. Hartigan, *Direct Clustering of a Data Matrix*, Journal of the American Statistical Association, 67:123–129,1972.

[2] I. S. Dhillon, S. Mallela and D. S. Modha, *Information-theoretic co-clustering*, KDD 2003

[3] M. M. Shafiei and E. E. Milios, *Latent Dirichlet Co-Clustering*, ICDM 2006.

[4] H. Shan and A. Banerjee, *Bayesian Co-clustering*, ICDM 2008.

[5] L. T. Watson, *Theory of globally convergent probability-one homotopies for non-linear programming*, SIAM Journal of Optimization, 2000.

[6] N. Tishby, F. C. Pereira and W. Bialek, *The information bottleneck method*, Allerton Conference on Communication, Control and Computing, 1999.

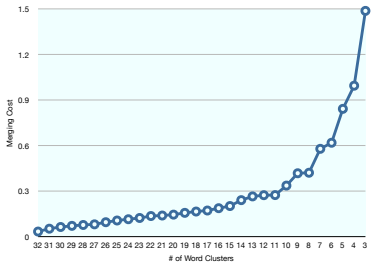


Figure 8: aIBCC: Merging Cost vs. # Word Clusters

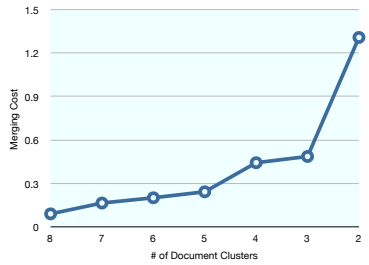


Figure 9: aIBCC: Merging Cost vs. # Document Clusters

[7] G. Elidan, N. Friedman and M. Chickering, *Learning hidden variable networks: The information bottleneck approach*, JMLR, 6:6-81, 2005.

[8] N. Friedman, O. Mosenzon, N. Slonim and N. Tishby, *Multivariate information bottleneck*, Neural Computation, 18:1739-1789, 2006.

[9] T. M. Cover and J. A. Thomas, *Elements of information theory*, Wiley-Interscience, 1991.

[10] K. Lang, *Newsweeder: Learning to filter netnews*, International Conference on Machine Learning, 1995.

[11] A. McCallum, *Simulated/Real/Aviation/Auto UseNet data*, http://cs.umass.edu/∼mccallum/code-data.html

[12] T. Mitchell, Machine Learning, McGraw Hill, 1997.

[13] N. Slonim, N. Friedman and N. Tishby, *Multivariate information bottleneck*, NIPS 2001.

[14] N. Slonim and N. Tishby, *Agglomerative information bottleneck*, NIPS 1999.

[15] G. Karypis, *Cluto - software for clustering high dimensional datasets*.

[16] A. Banerjee, I. Dhillon, J. Ghosh and S. Merugu, *A generalized maximum entropy approach to bregman co-clustering and matrix approximation*, KDD 2004.

[17] N. Slonim and N. Tishby, *Document clustering using word clusters via the information bottleneck method*, SIGIR 2000.

[18] R. El-Yaniv and O. Souroujon, *Iterative Double Clustering for Unsupervised And Semi-Supervised Learning*, NIPS 2001.

[19] N. Slonim, N. Friedman and N. Tishby, *Agglomerative Multivariate Information Bottleneck*, NIPS 2002.