Co-SOFT-Clustering: An Information Theoretic approach to obtain overlapping clusters from co-occurrence data

Swaminathan P Balaraman Ravindran
Department of Computer Science and Engineering
Indian Institute of Technology Madras
{swami,ravi}@cse.iitm.ac.in

Abstract

Co-clustering exploits co-occurrence information, from contingency tables to cluster both rows and columns simultaneously. It has been established that co-clustering produces a better clustering structure as compared to conventional methods of clustering. So far, co-clustering has only been used as a technique for producing hard clusters, which might be inadequate for applications such as document clustering. In this paper, we present an algorithm using the information theoretic approach [1] to generate overlapping (soft) clusters. The algorithm maintains probability membership for every instance to each of the possible clusters and iteratively tunes these membership values. The theoretical formulation of the criterion function is presented first, followed by the actual algorithm. We evaluate the algorithm over document/word co-occurrence information and present experimental results.

Introduction

Co-clustering is a relatively new clustering technique that looks at data contingency information (such as co-occurrence of documents/words, students/courses) to cluster both sides in an iterative fashion. For example, co-clustering can be used to cluster movies/viewers simultaneously in a movie database to produce clusters of similar movies and users with similar interests.

It has been established that clustering simultaneously on both sides gives an improved performance when compared to clustering either of the sides independent of the other [1]. So far, this technique has only been applied for producing hard clusters. Hard clustering algorithms place each data instance into exactly one cluster. This level of detail however, may not be sufficient in many cases such as in document clustering. We usually come across documents that discuss multiple, seemingly unrelated topics. In such cases, it becomes important from a classification perspective, to steer the document into multiple clusters, belonging to potentially different topics. We characterize multi-cluster membership of an instance by maintaining a probability distribution that describes its presence in each of the possible clusters. The guideline for clustering proposed in [1] is to minimize the loss in Mutual Information between the original row/column contingency distribution and the compressed distribution where multiple row (column) instances are clustered together. In this paper, we make use of the same guideline and present a different criterion function that can be used for a soft clustering task.

Theory

We shall denote the normalized co-occurrence matrix as the probability distribution P(x,y). Let there be k row clusters ($\hat{X} = \{\hat{x}_1, \hat{x}_2, \hat{x}_3, ..., \hat{x}_k\}$) and l column clusters ($\hat{Y} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_l\}$), without any loss of generality.

With every row x, we associate a vector C_k of size k, where $C_k^{\hat{x}}(x)$ refers to the probability of x belonging to cluster $\hat{x}$.

$$C_k^{\hat{x}}(x) = P(\hat{x} \mid x) \qquad \hat{x} \in \hat{X}$$

Similarly, we define an equivalent vector for each of the columns.

$$C_l^{\hat{y}}(y) = P(\hat{y} \mid y) \qquad \hat{y} \in \hat{Y}$$

These vectors are initialized to random values at the start of the algorithm. Our goal is to approximate the original distribution P(x,y) over the individual rows/columns to a compressed distribution over row/column clusters. We define the p.d.f in clustered space as follows.

$$P(\hat{x},\hat{y}) = \sum_x \sum_y P(\hat{x},\hat{y} \mid x,y)P(x,y) = \sum_x \sum_y P(\hat{x} \mid x,y)P(\hat{y} \mid x,y,\hat{x})P(x,y)$$

$$= \sum_x \sum_y P(\hat{x} \mid x)P(\hat{y} \mid y)P(x,y)$$

There is no significant information loss associated with dropping the conditional probabilities. Consider the term P($\hat{x} \mid x$). P($\hat{x} \mid x$) is calculated in the final algorithm as a function of y (and $\hat{y}$ in fact). This value of P($\hat{x} \mid x$) will be quite close to Mean(P($\hat{x} \mid x$,y)) over all y. Thus summing P($\hat{x} \mid x$,y) over all y is very much equivalent to summing P($\hat{x} \mid x$) (the mean) Y times. Therefore, we can get rid of the conditioning on y. In the second term, the conditioning on $\hat{x}$ is less informative in the presence of

(conditioning on) x. Hence, we can exclude $\hat{x}$. We can then use the same analogy as above to get rid of the conditioning on x. This can further be written in terms of our membership probabilities as follows.

$$P(\hat{x},\hat{y}) = \sum_{x}\sum_{y} C_k^{\hat{x}}(x) C_l^{\hat{y}}(y) P(x,y) \quad \text{--(1)}$$

The Mutual Information between x and y can be defined as

$$I(X;Y) = \sum_{x}\sum_{y} P(x,y)\log\frac{P(x,y)}{P(x)P(y)}$$

The Loss in MI due to compression (viz. Clustering) is given by,

$$\begin{aligned}
&I(X;Y) - I(\hat{X};\hat{Y}) \\
&= \sum_{x}\sum_{y} P(x,y)\log\frac{P(x,y)}{P(x)P(y)} - \\
&\sum_{\hat{x}}\sum_{\hat{y}}\sum_{x}\sum_{y} C_k^{\hat{x}}(x) C_l^{\hat{y}}(y) P(x,y)\log\frac{P(\hat{x},\hat{y})}{P(\hat{x})P(\hat{y})} \\
&= \sum_{\hat{x}}\sum_{\hat{y}}\sum_{x}\sum_{y} C_k^{\hat{x}}(x) C_l^{\hat{y}}(y) P(x,y)\left[\log\frac{P(x,y)}{P(\hat{x},\hat{y})\frac{P(x)}{P(\hat{x})}\frac{P(y)}{P(\hat{y})}}\right] \\
&= \sum_{\hat{x}}\sum_{\hat{y}}\sum_{x}\sum_{y} C_k^{\hat{x}}(x) C_l^{\hat{y}}(y) P(x,y)\left[\log\frac{P(x,y) C_k^{\hat{x}}(x) C_l^{\hat{y}}(y)}{P(\hat{x},\hat{y})P(x\mid\hat{x})P(y\mid\hat{y})}\right]
\end{aligned} \quad \text{--(2)}$$

We define the following distribution

$$q(x,y,\hat{x},\hat{y}) = P(\hat{x},\hat{y})P(x\mid\hat{x})P(y\mid\hat{y})$$

Now, (2) is the KL divergence

$$D\left(P(X,Y)C_k^{\hat{x}}(X)C_l^{\hat{y}}(Y) \,\|\, q(X,Y,\hat{X},\hat{Y})\right)$$

We shall state the following equalities, omitting proofs for brevity.

$$P(x) = q(x) \quad P(y) = q(y) \quad P(x,\hat{x}) = q(x,\hat{x}) \quad P(y,\hat{y}) = q(y,\hat{y})$$
$$P(x\mid\hat{x}) = q(x\mid\hat{x}) \quad P(\hat{y}\mid\hat{x}) = q(\hat{y}\mid\hat{x}) \quad P(\hat{x}\mid\hat{y}) = q(\hat{x}\mid\hat{y})$$

We now establish an important equality that helps us to represent the q distribution in an elegant form in order to further simplify (2).

$$q(y,\hat{y}\mid\hat{x}) = q(y\mid\hat{y})q(\hat{y}\mid\hat{x}) \quad \text{--(3)}$$

Thus, the q distribution can be written as

$$q(x,y,\hat{x},\hat{y}) = P(x)C_k^{\hat{x}}(x)q(\frac{y,\hat{y}}{\hat{x}}) \quad \text{--(4)}$$

(2) →

$$\begin{aligned}
&I(X;Y) - I(\hat{X};\hat{Y}) \\
&= \sum_{\hat{x}}\sum_{\hat{y}}\sum_{x}\sum_{y} P(x,y)C_k^{\hat{x}}(x)C_l^{\hat{y}}(y)\log\frac{P(x,y)C_k^{\hat{x}}(x)C_l^{\hat{y}}(y)}{q(x,y,\hat{x},\hat{y})} \\
&= \sum_{\hat{x}}\sum_{x} P(x)C_k^{\hat{x}}(x) D\left(P(Y\mid x)C_l^{\hat{y}}(Y) \,\|\, q(\frac{Y,\hat{Y}}{\hat{x}})\right)
\end{aligned} \quad \text{--(5)}$$

(5) leads to the final algorithm. Since our goal is to minimize the overall loss in Mutual Information, we should direct each row x, into the cluster $\hat{x}$ for which the KL Divergence $D[P(Y|x)C_l^{\hat{Y}}(Y) \,\|\, q(Y,\hat{Y}|\hat{x})]$ is minimum. We start the algorithm by randomly initializing the membership co-efficients (ensuring the probability constraints). It is important to note that the initial values should not be made perfectly uniform, as this would be a fixed point from which no update would be possible. We evaluate the required $P(\hat{x},\hat{y})$ and q distributions and then for each row x, we decrement its membership to cluster $\hat{x}$ by α times the KL Divergence corresponding to cluster $\hat{x}$, where α is the learning rate parameter. We repeat this step for every possible row cluster. It is easy to see that the cluster that best fits x will get the least decrement value (of the corresponding membership co-efficient). We repeat the above procedure for columns and iterate till convergence.

Experimental Results

We chose various subsets of the 20-newsgroup dataset from both unclean and cleaned-up (duplicates and unwanted headers removed) versions. The experimental procedure is to first run a hard clustering algorithm (say *k-means*) on the dataset, and mark those documents that fall close to two or more cluster centroids, i.e. choose those documents for which the difference in distance to two of the nearest centroids is below a certain threshold. We term these documents to be potentially *soft-clusterable.* We then run co-SOFT-clustering on the dataset, pick those documents that are soft clustered and look at correspondence with *k-means* soft-clusterables. The documents that are soft clustered by either algorithm are manually labeled (two experts, χ^2 33.278, sig 0.001). We ran experiments with different settings for word/document cluster count, and achieved optimum results for values k=30, l=12 on subsets of 20-newsgroups with about 300-500 documents. While clustering datasets with mostly soft documents, we achieved up to 95% agreement with k-means and 80% with manual labeling. In case of datasets with diverse documents, we had very low agreement (0%-19%) with k-means (this reflects the true nature of the dataset, as most of the documents are not soft), and high agreement (76%-91%) with manual labeling.