Soft Co-Clucstering Via Extension of PLSA

Dakan Wang

1 introduction

In recent years, co-clustering has emerged to be a powerful data mining tool for two-dimensional co-occurrence and dyadic real data. Typical examples include co-clustering word-document matrix, user-query matrix, etc. Co-clustering algorithms always outperform the traditional one way clustering in problems with sparse and high-dimensional co-occurrence data, since they simultaneously cluster rows and columns of a matrix. Co-clustering has been applied to recommendation systems, microarray analysis, etc.

There have been already many works on algorithms for co-clustering [1, 2, 3]. [1] proposed an information theoretical co-clustering algorithm. However, their method is under the hard setting, i.e., it just allows one row or column to belong to only one cluster. Such restrictions are not suitable in real world applications. In real life, one object can belong to more than one categories, and a soft co-clustering framework allowing mixed membership may make more sense. It is worth pointing out that [3] also proposed a soft co-clustering model. However, their model is a bit complex. We will propose a very simple and intuitive model.

We would like to extend the PLSA to a soft co-clustering model which allows that both columns and rows can belong to different clusters. The idea is that we add two latent sets of nodes to the model. The two latent sets correspond to the clusters for rows and columns correspondingly. The training of the model is similar to PLSA and is very efficient and easy to implement.

2 Model

In this section, we will describe the 2-layer Probabilistic Latent Semantic Analysis(2PLSA) and a corresponding EM algorithm.

Following the idea of PLSA, we assume that the probability of a word w given a document d is defined as follows

$$p(w|d) = \sum_{z \in Z} \sum_{s \in S} p(w|z)p(z|s)p(s|d) \tag{1}$$

Here z is the word cluster, and s is the document cluster. This equation can be interpreted as follows

- Pick a document topic s from d with probability $p(s|d)$

- Pick a word topic z from s with probability $p(z|s)$
- Generate a word w from word topic z with probability $p(w|z)$

From the above formulation, we can write the likelihood function to be

$$\begin{aligned} L &= \sum_{i,j} n(d_i, w_j) \log p(w_j|d_i)p(d_i) \\ &= \sum_{i,j} n(d_i, w_j) \log \sum_{k,l} p(w_j|z_k)p(z_k|s_l)p(s_l|d_i)p(d_i) \\ &= \sum_{i,j} n(d_i, w_j) \log \sum_{k,l} q(z_k, s_l|d_i, w_j) \frac{p(w_j|z_k)p(z_k|s_l)p(s_l|d_i)p(d_i)}{q(z_k, s_l|d_i, w_j)} \end{aligned}$$

3 inference

We briefly summarize the EM algorithm here. Detailed derivations are similar to those instructed in the class.

3.1 E-Step

$$q(s_l, z_k|d_i, w_j) = \frac{p(w_j|z_k)p(z_k|s_l)p(s_l|d_i)}{\sum_{k,l} p(w_j|z_k)p(z_k|s_l)p(s_l|d_i)} \tag{2}$$

3.2 M-step

$$\begin{aligned} p(s_l, z_k, d_i, w_j) &= q(s_l, z_k|d_i, w_j)p(d_i, w_j) \\ p(w_j|z_k) &= \frac{\sum_{i,l} p(s_l, z_k, d_i, w_j)}{\sum_{i,j,l} p(s_l, z_k, d_i, w_j)} \\ p(z_k|s_l) &= \frac{\sum_{i,j} p(s_l, z_k, d_i, w_j)}{\sum_{i,j,k} p(s_l, z_k, d_i, w_j)} \\ p(s_l|d_i) &= \frac{\sum_{j,k} p(s_l, z_k, d_i, w_j)}{\sum_{j,k,l} p(s_l, z_k, d_i, w_j)} \end{aligned} \tag{3}$$

4 Equivalence with information theoretical co-clustering

[1] defined a co-clustering algorithm from information theory perspective. If x and y are respectively clustered into $\widehat{x}$ and $\widehat{y}$, the loss in mutual information is

$$I(X;Y) - I(\widehat{X};\widehat{Y}) = D(p(X,Y)||q(X,Y)) \tag{4}$$

Here $D(\cdot||\cdot)$ is the KL divergence, and $q(X,Y)$ is defined to be

$$q(x,y) = p(\widehat{x}.\widehat{y})p(x|\widehat{x})p(y|\widehat{y}) \tag{5}$$

where

$$p(\widehat{x},\widehat{y}) = \sum_{x\in\widehat{x}}\sum_{y\in\widehat{y}} p(x,y)$$
$$p(x|\widehat{x}) = \frac{p(x)}{p(\widehat{x})}\ p(y|\widehat{y}) = \frac{p(y)}{p(\widehat{y})} \tag{6}$$

Now we prove the equivalence of our model and that in [1] under the hard setting. We rewrite the likelihood function of our model to be

$$\begin{aligned} L &= \sum_{i,j} n(d_i,w_j) \sum_{k,l} \log p(w_j|z_k)p(z_k|s_l)p(s_l|d_i)p(d_i) \\ &= \sum_{i,j} n(d_i,w_j) \log \sum_{k,l} p(w_j|z_k)\frac{p(z_k,z_l)}{p(s_l)}\frac{p(d_i|s_l)p(s_l)}{p(d_i)}p(d_i) \\ &= \sum_{i,j} n(d_i,w_j) \log \sum_{k,l} p(w_j|z_k)p(z_k,s_l)p(d_i|s_l) \end{aligned} \tag{7}$$

If we restrict the above equation to hard setting, we get

$$L = \sum_k \sum_l \sum_{d_i\in s_l, w_j\in z_k} n(d_i,w_j) \log p(w_j|z_k)p(z_k,s_l)p(d_i|z_l) \tag{8}$$

On the other side, in information theoretical co-clustering [1], we want to minimize

$$\begin{aligned} & I(D,W) - I(S,Z) \\ = & \sum_k \sum_l \sum_{d_i\in s_l, w_j\in z_k} p(d_i,w_j)\log p(d_i,w_j) \\ - & \sum_k \sum_l \sum_{d_i\in s_l, w_j\in z_k} p(d_i,w_j)\log p(w_j|z_k)p(z_k,s_l)p(d_i|s_l) \end{aligned} \tag{9}$$

Notice that tbe first term is constant given a collection of documents and words, thus we only need to maximize

$$\sum_k \sum_l \sum_{d_i\in s_l, w_j\in z_k} p(d_i,w_j)\log p(w_j|z_k)p(z_k,s_l)p(d_i|s_l) \tag{10}$$

which differs from our maximum likelihood formulation by only a constant scalar.

5 Experiments

5.1 Datasets

In order to evaluate our algorithm performance more extensively, we conducted experiments using two datasets from different domains. The first domain is

Tab. 1: Extracted Topics

1	2	3	4	5
image	drive	god	israel	space
graphical	scsi	think	jew	nassa
file	mb	believe	arab	orbit
program	disk	christ	war	launch
fotmat	control	athetist	kill	earth

the 20 Newsgroups dataset, which is a text collection of approximately 20,000 newsgroup documents across 20 different newsgroups . Another dataset comes from the Wikipedia XML dataset, from which we had sampled a subset of 9 document topics composing a total of 1,236 articles as what was done in [2]. Some preprocessing has been applied to on the raw text data. We had firstly performed the Porter stemmer on terms and removed all the stop words from a stop word list.

5.2 Word and Document Clusters

Our algorithm could return document and word clusters easily by doing a sorting on the corresponding $p(w|z)$ and $p(d|s)$. In this section, we would list the 5 most possible words in 5 topics from the 20 Newsgroup dataset, such a table can be directly constructed by sorting $p(w|z)$. The results are shown in Table 1. From Table 1 and the ground truth labeling, we could see that topic 1 to topic 5 will correspond to comp.graphics, comp.sys.ibm.pc.hardware, talk.religion.misc, talk.politics.mideast and sci.space. We could observe that the word clusters we got are meaningful and representative of different topics.

5.3 Document Clustering Performance

We compared our proposed 2PLSA in terms of clustering accuracy with two baselines, ITCC and LDCC. We conducted our experiment on the Wikipedia XML corpus to directly compare our model with the results in [2]. The metrics we used are Precision, Recall and F1-Measure calculated over pairs of points. In our co-clustering model, for two documents d_1 and d_2, we determine whether they belong to the same class by calculating $\sum_s p(s|d_1)p(s|d_2)$ and if the value is larger than 0.5, we consider d1 and d2 to belong to the same category. As [2], we also tested the performance of our algorithm with the number of topics set to be 15 and 20.

Table 2 shows the result of comparisons. It shows that our algorithm could significantly outperform the baseline methods, in terms of precision and F1-Score. We also would like to add a side note that since our algorithm is based on the more efficient PLSA model, the computational complexity of our algorithm is significantly lower than that of LDCC.

Tab. 2: Performance Comparison

Algorithm	S	Precision	Recall	F1-Score
2PLSA	15	38.56	70.38	48.83
LDCC	15	30.88	79.21	44.43
ITCC	15	31.06	7.44	12.00
2PLSA	20	37.32	70.17	48.75
LDCC	20	31.10	75.50	44.09
ITCC	20	31.06	6.16	10.28

Finally, we show the rearranged co-occurrence matrix after co-clustering. We sampled a subset of 10 topics from 20 Newsgroup dataset with 4,980 documents and 24,864 words. The left subfigure of Figure 1 shows the original word-document matrix and the right subfigure shows the re-ordered matrix by arranging rows and columns according to the document cluster order and the word cluster order, respectively. Note that in the right subfigure, the clusters are not falling around the main diagonal since that requires doing topic alignment between word topics and document topics.

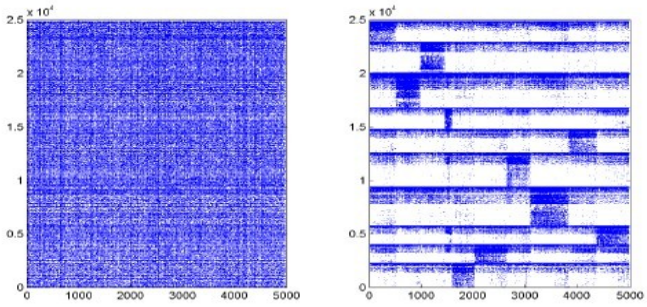

Fig. 1: co-occurrence matrix before and after co-clustering

References

[1] Inderjit S. Dhillon, Subramanyam Mallela, and Dharmendra S. Modha. Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 89–98, New York, NY, USA, 2003. ACM.

[2] M.M. Shafiei and E.E. Milios. Latent dirichlet co-clustering. In *Data Mining, 2006. ICDM '06. Sixth International Conference on*, pages 542 –551, 2006.

[3] Hanhuai Shan and Arindam Banerjee. Bayesian co-clustering. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 530–539, Washington, DC, USA, 2008. IEEE Computer Society.