Clustering via Content-Augmented Stochastic Blockmodels

Massey Cashore, Peter I. Frazier, Xiaoting Zhao, Yujia Liu

October 12, 2014

Abstract

A large portion of the data being created on the web contains interactions between users and items. Most research in this domain has been concerned with recommender systems, that is, determining items to show users to maximize some value associated with the interactions. Here we consider the problem of clustering the items, that is, partitioning them in a manner that coincides with their structure. We first note that by considering user data, stochastic blockmodels can be easily adapted to find a meaningful clsutering of the documents. We then modify the assumptions of a stochastic blockmodel to incorporate item content. Doing so unifies content-based and collaborative filtering, the two main approaches to recommender systems. We see improvement in the content-augmented clusters over the purely stochastic blockmodel clusters, as well as other state-of-the-art clustering algorithms, with respect to various metrics.

1 Introduction

Stochastic blockmodels were introduced to discover latent community structure in graphs (Holland et al., 1983), typically formed by people or other entities interacting with each other (each person is a nodes, and edges indicate interactions), or by people interacting with text documents, images, videos, or some other object (each person is a node, each object is a node, and edges indicate interactions, forming a bipartite graph). In this second kind of application, interaction information is the only information typically used, and information from the documents themselves is ignored.

In this paper, we provide an algorithm that uses both the interaction information, **and document contents**, to detect communities more reliably. As far as we are aware, ours is the first model to use both kinds of information in a stochastic blockmodel community-detection framework.

Traditional bipartite stochastic blockmodels assume that different communities tend to interact differently with each text documents, with some communities tending to interact more frequently with a given document type, and other communities interacting more frequently with other document types. This differential preference of communities for documents induces a latent document clustering, which bipartite stochastic blockmodels attempting to learn this latent clustering from interaction information alone. Our model adds an additional assumption: that documents in each cluster have distinct characteristics, observable in the words that occur in them. When this assumption is satisfied, we argue that it can and should be used to improve performance.

While this assumption does not necessarily hold in all community detection applications, we argue that it holds in a wide variety of settings. In this paper, we apply this model to scientists interacting with scientific articles, and to political blogs linking to each other [pf: can we apply this to a political blogs dataset?], where the words that tend to appear in articles preferred by a community vary considerably from community to community. Our model could also be applied to communities interacting with other kinds of items, e.g., videos, but in our empirical studies we focus on text.

Our model can also be seen as a co-clustering algorithm, because it provides a clustering of both users and documents. However, our model is distinguished from all other co-clustering algorithms of which we are aware, in that it uses not just the interaction information, but also co-variates observable in the documents. Thus, our model is distinguished from co-clustering approaches that use only document comments (e.g., based only the matrix of word co-occurrence, such as (cites)) by the way it takes advantage of user co-access to find the mapping of contents to clusters that matches the communities' preferences. It is distinguished from co-clustering approaches that only use user interactions (e.g., cites) in that document contents are used to refine and improve the co-clustering.

An additional advantage of including document covariates into our model is new documents with no interaction history can be included into an appropriate document cluster, addressing the cold-start problem.

In the context of recommender systems, there has been growing interest in combining user interaction data (used by collaborative filtering (Su and Khoshgoftaar, 2009; Adomavicius and Tuzhilin, 2005)), with item content (used by content-based filtering (Pazzani and Billsus, 2007; Manning et al., 2008)) to provide improved clustering and improved recommendation (Wang and Blei, 2011; Claypool et al., 1999; Balabanović and Shoham, 1997; Salter and Antonopoulos, 2006; Basilico and Hofmann, 2004; Melville et al., 2002). (also cite Laurent's poisson factorization paper.) Our paper can be seen as as adding to this growing literature, but focusing on community detection and co-clustering, rather than recommendation.

In section 2 we provide a detailed description of the model, and in section 3 we construct a Gibbs sampler which can be used to perform inference. In section 4 we apply the model to three real-world datasets, where we compare our clusters to those obtained using stochastic blockmodels that ignore item content, as well as more traditional clusters obtained using kmeans on vector representations of the items. We find our model provides a nice balance between item and user generalizability, and superior performance with respect to partitioning user interests. Additionally, we find our clusters have the ability to accurately partition items with respect to a phenomenon where papers on arXiv.org (arXiv.org, 2014), were being submitted to the wrong subcategory.

2 Content-Augmented Stochastic Blockmodels

Suppose we are dealing with an application on the web such that there are D items and U users potentially interested in the items. Suppose that over time the users have been shown and provided feedback on a subset items. We encode their feedback with the variable Y, defined by

$$Y_{ij} = \begin{cases} 1 & \text{if item } i \text{ was shown to user } j \text{ and deemed relevant} \\ 0 & \text{if item } i \text{ was shown to user } j \text{ and deemed irrelevant} \\ \triangle & \text{if item } i \text{ was not shown to user } j. \end{cases}$$

Note we only consider a binary response (relevant / irrelevant), and use the symbol $\triangle = Y_{ij}$ to denote the case where item i is not shown to user j.

The model proceeds by assuming there are k_1 clusters such that each item i belongs to cluster $z_i \in \{1, \ldots, k_1\}$ and there are k_2 communities such that each user j belongs to community $w_j \in \{1, \ldots, k_2\}$. We assume the community membership of a user and cluster membership of a paper completely determines the probability the user finds the paper relevant. Explicitly, the model assumes that

$$p(Y_{ij} = 1 | z_i = x, w_j = y, Y_{ij} \neq \triangle) = q_{xy}$$

for constants q_{xy} ranging over item clusters x and user communities y. For simplicity, we encode the q_{xy} as a matrix $Q = [q_{xy}]$. We assume the observed Y_{ij} are all sampled independently.

Finally, we endow the latent variables described above with the following Bayesian priors:

- The cluster z_i of item i follows a uniform distribution on k_1 elements.
- The community w_j of user j follows a uniform distribution on k_2 elements.
- The cluster-community interest probabilities q_{xy} follow a Beta(α, β) for some $\alpha, \beta > 0$.

The model described up to this point is the stochastic blockmodel for clustering that ignores item content. In section 4 we will use this model as a benchmark, and refer to the resulting clusters as stochastic blockmodel (SB) clusters.

In order to augment the model with content, we suppose that each item can be represented by an F-dimensional feature vector, such that the nth entry counts how many time the nth trait occurred, for some set of F traits. This representation is inspired by the case where items are scientific papers and the traits are clusters of similar words, as is described on in the following section. Let d_i represent the feature vector for the ith item.

In order to force that items in the same cluster should be similar in content, we assume that associated with each item cluster x is a probability vector $p_x \in [0,1]^F$, $\sum_\ell p_{x\ell} = 1$, such that if item i is in cluster x

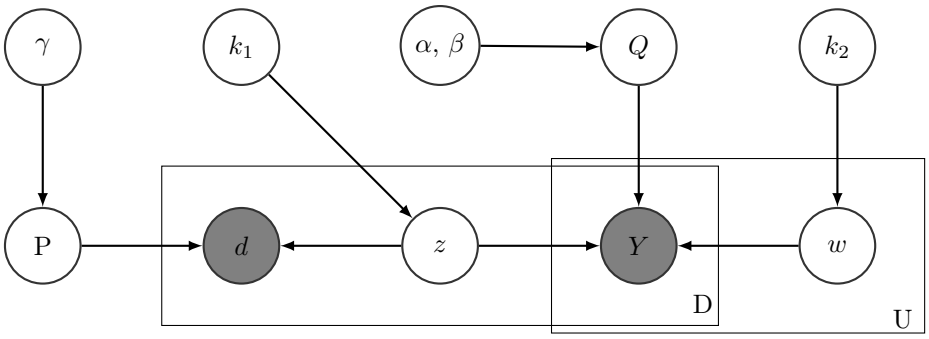

Figure 1: Graphical representation of the content-augmented model

then the feature vector d_i is created from N_i samples of a Multinomial(p_x) distribution (the process by which N_i is chosen is unimportant for the model). We assume the observed d_i are all sampled independently, and we place a Dirichlet(γ) prior on each p_x, for some $\gamma \in (\mathbb{R}_{>0})^F$.

This fully describes the content-augmented stochastic blockmodel (CASB). Standard inference techniques can be used to estimate all latent variables, and in the following section we explicitly construct a Gibbs sampler for this purpose. A graphical depiction of the CASB can be seen in Figure 1, illustrating the dependencies between all latent variables.

3 Inference

To perform inference on the CASB, we construct a Gibbs sampler to obtain samples from the posterior distribution, taking the sample with the highest likelihood as our estimate. Gibbs samplers work by holding each variable in the distribution fixed save one, and sampling from that simpler conditional distribution to obtain a new estimate for the non-fixed variable. Thus in order to construct the Gibbs sampler we must describe the complete conditional distributions for each of the latent variables. To do so, we start by determining the likelihood function. Since the observed Y_{ij} are independent, and the d_i are independent, by conditioning on all the unobserved data we can write down the likelihood function

$$L(z,w,P,Q|d,Y) = p(d,Y|z,w,P,Q) \tag{3.1}$$

$$= \left(\prod_{i=1}^{D} p(d_i|z_i,p_{z_i})\right)\left(\prod_{(i,j):Y_{ij}\neq\triangle} p(Y_{ij}|z_i,w_j,q_{z_iw_j})\right) \tag{3.2}$$

$$\propto \left(\prod_{i=1}^{D}\prod_{\ell=1}^{F} p_{z_i\ell}^{di\ell}\right)\left(\prod_{(i,j):Y_{ij}=1} q_{z_iw_j}\right)\left(\prod_{(i,j):Y_{ij}=0} 1-q_{z_iw_j}\right). \tag{3.3}$$

Knowledge of the likelihood function, as well as the priors on each of the latent variables, gives us knowledge of the posterior distribution:

$$p(z,w,P,Q|d,Y) \propto L(z,w,P,Q|d,Y)p(z,w,P,Q) \tag{3.4}$$

$$= L(z,w,P,Q|d,Y)p(z)p(w)p(P)p(Q). \tag{3.5}$$

The complete conditionals for z and w are easiest, so we describe these first. Since we have a uniform discrete prior on each z_i and w_j, it follows that $p(z) = \left(\frac{1}{k_1}\right)^D$ which, in fact, does not depend on z. Thus we can absorb $p(z)$ along with $p(w), p(P), p(Q)$ into a constant, and conclude $p(z_i|z_{-i},w,P,Q,n,Y) \propto L(z = (z_{-i},z_i),w,P,Q|n,Y)$. Considering only the terms in (3.3) that depend on z_i, we conclude

$$p(z_i|z_{-i},w,P,Q,d,Y) \propto \left(\prod_{\ell=1}^{F} p_{z_i\ell}^{d_{i\ell}}\right)\left(\prod_{j:Y_{ij}=1} q_{z_iw_j}\right)\left(\prod_{j:Y_{ij}=0} 1-q_{z_iw_j}\right).$$

In a similar fashion, we conclude the complete conditional for w_j is

$$p(w_j|w_{-j}, z, P, Q, d, Y) \propto \left(\prod_{i:Y_{ij}=1} q_{z_i w_j}\right)\left(\prod_{i:Y_{ij}\neq 1} 1 - q_{z_i w_j}\right).$$

Sampling from these distributions amounts to sampling from a multinomial with a known parameter.

The full conditionals for Q and P are not as simple, but work out nicely since we endowed them with conjugate priors. Absorbing the terms in (3.5) that do not depend on q_{xy} into a constant, we conclude

$$p(q_{xy}|Q_{-xy}, w, z, P, d, Y) \propto \left(\prod_{(i,j):z_i=x,w_j=y,Y_{ij}=1} q_{xy}\right)\left(\prod_{(i,j):,z_i=x,w_j=y,Y_{ij}=0} 1 - q_{xy}\right) p(q_{xy}).$$

For item cluster x and community y recall the prior on q_{xy} is Beta(α, β). Thus we know $p(q_{xy}) \propto q_{xy}^{\alpha-1}(1-q_{xy})^{\beta-1}$. Hence it follows that

$$q_{xy}|Q_{-xy}, w, z, P, d, Y \sim \text{Beta}\left(\alpha + \sum_{(i,j):z_i=x,w_j=y} Y_{ij}, \beta + \sum_{(i,j):z_i=x,w_j=y} 1 - Y_{ij}\right).$$

Finally, we describe the complete conditional for P. Since our prior is a Dirichlet(γ) distribution on the entire vector p_x, we describe the complete conditional distribution for the entire vector, as opposed to the individual elements $p_{x\ell}$. Proceeding as usual by absorbing the terms not depending on p_x from (3.3) into a constant, we have

$$p(p_x|P_{-x}, w, z, Q, d, Y) \propto \left(\prod_{i:z_i=x}\prod_{\ell=1}^{F} p_{x\ell}^{d_{i\ell}}\right) p(p_x).$$

However, since $p_x \sim$ Dirichlet(γ), we have $p(p_x) \propto \prod_{\ell=1}^{F} p_{x\ell}^{\gamma_\ell - 1}$. Thus it follows that

$$p_x|P_{-x}, w, z, Q, d, Y \sim \text{Dirichlet}(\gamma')$$

where $\gamma'_\ell = \gamma_\ell + \sum_{i:z_i=x} d_{i\ell}$.

Having describe the four necessary conditional distributions, a Gibbs sampler may be implemented to obtain samples from the posterior distribution. Fitting the model to a legitimate dataset then amounts to running the Gibbs sampler until the samples converge, and choosing the estimate with the highest likelihood.

4 Experimental Results

We performed experiments on three real-world datasets, two taken from arXiv.org and one taken from the collection of talks given at the annual INFORMS meeting.

The first of the arXiv datasets consisted of 10861 documents from the cosmology subcategory (astro-ph.CO) posted between 2009 and 2010, along with 500 arbitrarily selected users with rich history in this subcategory. The second arXiv dataset was a medley of documents from this subcategory, as well as the galaxies (astro-ph.GA) and earth and planetary astrophysics (astro-ph.EP) subcategories, along with 846 users with rich history in at least one of these subcategories.

The third dataset, illustrating the flexibility of the model, was taken from the presentations given at the annual INFORMS meeting. Presentations at this meeting occur in sessions, where sessions are small collections of similar presentations. We consider invited and sponsored sessions, where the talks in a given session are determined by that session's chair. We hypothesize that this method of assigning talks to sessions is more indicative of the underlying community structure, in contrast to situations where one organizer determines all sessions. We used 450 speakers as the users, such that a speaker is interested in a talk if the talk under consideration and the speaker's own talk occured in the same session. Note that with this dataset, a speaker is always either interested or disinterested in a talk (i.e. $Y_{ij} \neq \triangle$ for all i, j).

As mentioned earlier, the item representations described earlier are directly motivated by the case where items are text documents. The item representations in this case are then taken by applying word2vec (Mikolov

Titles from non-empty cluster 1 (experimental)	Titles from non-empty cluster 2 (theoretical)
Optical variability of radio-intermediate quasars	Gravitational-Wave Stochastic Background from Kinks and Cusps on Cosmic String
A study of the interplay between ionized gas and star clusters in the central region of NGC 5253 with 2D spectroscopy	Large non-Gaussianities in the Effective Field Theory Approach to Single-Field Inflation: the Bispectrum
The dark and dusty side of galaxy evolution	Dwarf Galaxies, MOND, and Relativistic Gravitation

Table 1: Titles from 3 arbitrarily chosen papers from each of the nonempty clusters

et al., 2013) to each of the datasets, resulting in each word being associated with a high-dimensional vector such that words with small cosine distance are semantically similar. We then used kmeans to cluster these words. The feature vector d_i associated with each item i then become counts of how many times words from each cluster appeared. That is, $d_{i\ell}$ represents how many times words from the ℓth word cluster appeared in the ith document.[1]

For the arXiv datasets, we ran word2vec on the paper contents and generated 1024 word clusters. For the INFORMS data we ran word2vec on the abstracts associated with each talk. We generated 100 word clusters for the INFORMS data as the total number of words was significantly smaller.

In the following subsections we provide the details and results of several experiments comparing the CASB clusters to the simpler SB clusters, as well as clusters formed by running kmeans on the item representations. One of these experiments required fitting the CASB clusters to the cosmology dataset with $k_1 = k_2 = 2$. Table 1 provides an emprical evaluation of these clusters, showing three arbitrarily selected papers from each. The tables suggests one cluster contains experimental papers, while the other contains theoretical papers.

A common method of evaluating how well a statistical model fits a dataset is to withold a portion of the data when the model is fit, and see how well the learned parameters explain the held out data. For the datasets under our consideration there are two components to consider: user behaviour and item content.

4.1 Generalizing User Behaviour To evaluate the generalizability of the model with respect to user behaviour, we held out 1000 documents from each of the three datasets and fit the CASB, setting $k_1 = k_2 = 5$. As a benchmark, we fit the simpler SB model with $k_1 = k_2 = 5$, obtaining purely collaborative clusters (no kmeans clusters were used in this experiment, as they ignore user information).

To calculate the unexpectedness of the user click data, for each held out document t we formed the vector $q_t \in \mathbb{R}^{k_2}$ where q_{ty} represents the fraction of users in community y who clicked the document, over the number of users who were shown the document. This vector then lies in the same space as the vectors $q_x = [q_{xy}]_{1 \le y \le k_2}$ representing the probabilities of a user in community y clicking on a document from cluster x, as specified by the model. Thus the cluster that best matches the dth held out document based on user interactions is the cluster x that minimizes $\|q_x - q_t\|$. Let the error for the tth held out document be defined by $e_t = \|q_x - q_t\|$ where x is the best cluster. We can then measure the unexpectedness of the user interactions for the entire dataset as the average error, $\sum_t e_t/1000$.

Table 2 shows the average error for both models for each dataset. A priori, one would expect the SB model to perform better by this measure, as the entire clustering scheme is based solely off user interaction data. The CASB model is forced to account for the document representations as well, hence placing less weight on the user interactions. As expected, the SB clusters generally have lower errors. With the medley dataset the CASB clusters have poor generalizability, with an error term double that of the SB clusters. However, for the cosmology dataset, the CASB error term is much closer to that of the SB error term, and for the INFORMS dataset the terms are equal.

These results suggest that the generalizability of the models to user interactions is highly dependent on the dataset. One possible explanation could be that the medley dataset contains users and documents from three different subcategories, fragmenting the connections between users and papers. This is supported by

[1] We would like to thank Alexander Alemi for the idea of the word2vec representations, as well as providing them to us for the arXiv data.

	CASB	SB
cosmology	0.058	0.046
medley	0.048	0.025
INFORMS	0.021	0.021

Table 2: User generalization error for each model with respect to the three datasets

	CASB	SB	kmeans
cosmology	1111.53	1196.89	803.83
medley	1229.61	1317.83	744.07
INFORMS	8.09	8.43	8.06

Table 3: Document generalization error for each model with respect to the three datasets

the fact that the best generalizability is achieved with the INFORMS dataset, where there is a connection between all pairs of users and items.

4.2 Generalizing Item Content The generalizability of the models to item content can be measured in a similar manner to above. We associated the vector d_x with each cluster x, where d_x is the average of all document vectors d_i such that document i belongs to cluster x. The tth held out document was then assigned to the cluster x minimizing $\|d_x - d_t\|$, calling this error term e_t . We again define the average error for the dataset as the error of the e_t.

Table 3 summarizes the content errors for the three datasets with respect to all clustering schemes. Note that kmeans by far has the best performance. However, the entire premise of kmeans is to minimize the distance of the vectors being clustered to the centre of the cluster, which is exactly the metric under consideration. Considering only the CASB and SB clusters, we see that CASB clusters outperform the generalization capabilities of SB clusters with respect to item content. This is expected behaviour, as the CASB explicitly considers item content, while the SB does not.

Note that the magnitude of the errors with respect to the medley dataset is largest for the CASB and SB clusters, and the gap between CASB and SB generalizability is relatively small. On the other hand, the error term associated with the INFORMS clusters are small for all clusterings, with the CASB error term being only marginally larger than the SB clusters. (Note the large discrepancy in magnitude between the arXiv error terms and the INFORMS error terms is due to the INFORMS abstract representations living in $\mathbb{R}^{100}$ while the arXiv document representations lie in $\mathbb{R}^{1024}$.)

These results are consistent with the previous section, where the worst generalizability was seen with the medley dataset, and the best was seen with the INFORMS dataset. Again, this suggests CASB model fit is highly dependent on the dataset at hand.

4.3 Partitioning User Interests One of the motivating examples for a hard clustering of items given in the introduction was that of a content browser, allowing users of a system to browse a series of otherwise unstructured content in a principled manner. A desirable property for such a browser would be partitioning user interests, meaning that a user would only need to peruse a subset of the clusters in order to be presented with everything relevant to them.

To test this property, we held out 100 users from each of the datasets and again fit both models and computed the kmeans clusters. For each held out user u, we formed the vector $\vec{u} = [u_x]_{1 \leq x \leq k_1}$ where u_x is the fraction of items the user was interested in that fell into cluster x. Let $\vec{v}_u$ be the result of sorting $\vec{u}$ in descending order. If the uth user has highly partitioned interests, then the first few values of $\vec{v}_u$ will nearly sum to 1, while the remaining values will be negligible. We can then characterize how well each clustering partitions user interests in each dataset by the vector resulting from averaging the collection of $\vec{v}_u$.

For each dataset, Figure 2 plots the partition vectors for each clustering. Surprisingly, for the cosmology dataset kmeans clusters are more successful at partitioning user interests than both CASB and SB clusters,

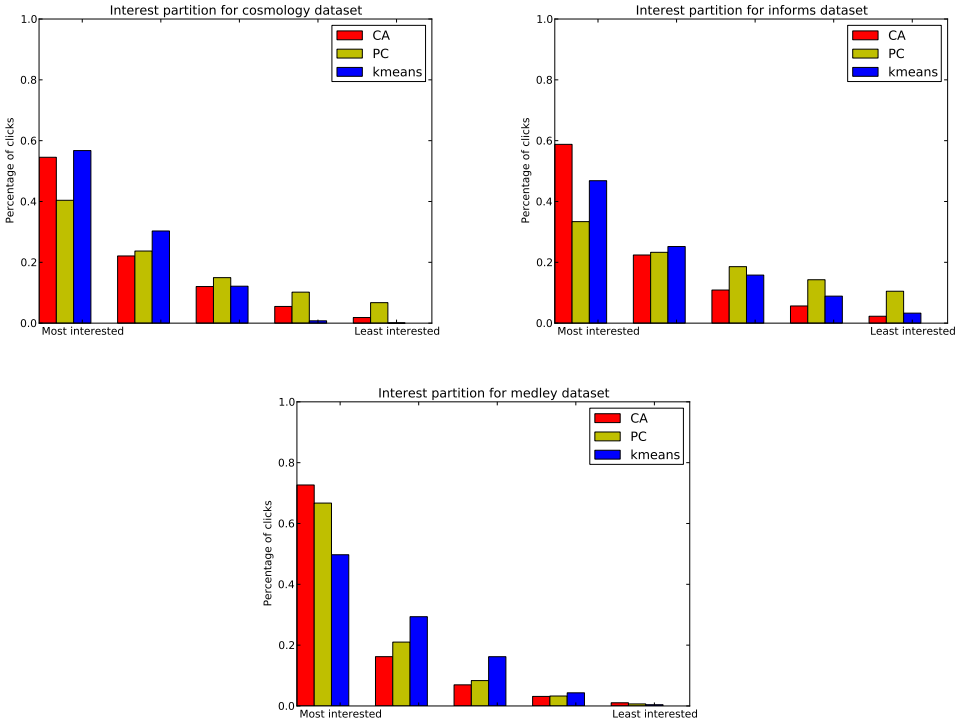

Figure 2: Partition of user interests for the three datasets

though only marginally beating CASB. Also surprising is that SB clusters do not generally provide well-partitioned user interests, despite their inherent collaborative nature. On the other hand, CASB clusters provide highly favourable interest partitions for the INFORMS and medley datasets, while being beaten out only slightly in the cosmology dataset.

Interestingly, the CASB and SB clusters do the best job of partitioning interests in the medley dataset, the one with which they had poorest generalizability.

4.4 Capturing Misplaced Papers This evaluation focuses on two astrophysics subcategories on the arXiv: Cosmology and Nongalactic Astrophysics (astro-ph.CO); and Astrophysics of Galaxies (astro-ph.GA).

In creating these categories, the arXiv administrators' intention was for all papers about galactic astrophysics to go to astro-ph.GA. However, in the past, a significant portion of the astrophysics community had a different interpretation: astro-ph.GA was for papers discussing our galaxy, the Milky Way, while papers discussing other galaxies should go to astro-ph.CO.

In late 2013, arXiv.org's moderators began enforcing their interpretation of these two subcategories, recategorizing papers about galaxies from astro-ph.CO to astro-ph.GA. (Ginsparg, 2014).

We hypothesized that the research communities interested in nongalactic and galactic papers differ, as do the words in their papers, and so the content-augmented stochastic blockmodel should be able to separate older papers from astro-ph.CO into those nongalactic papers that were correctly submitted to astro-ph.CO, and those galactic papers that should have been submitted to astro-ph.GA. Moreover, it should be able to do this in an unsupervised way, based only on usage and item content, without being given examples of papers in each class.

To test this hypothesis, we fit the CASB to papers submitted to astro-ph.CO over 2009-2010, setting $k_1 = k_2 = 2$. As baselines, we used the SB with $k_1 = k_2 = 2$, and kmeans clustering with $k = 2$. We then compared each of these clusterings to a ground truth classification of papers (described below) into those that were properly submitted to astro-ph.CO, and those that should have been submitted to astro-ph.GA. The results are pictured below in Figure 3.

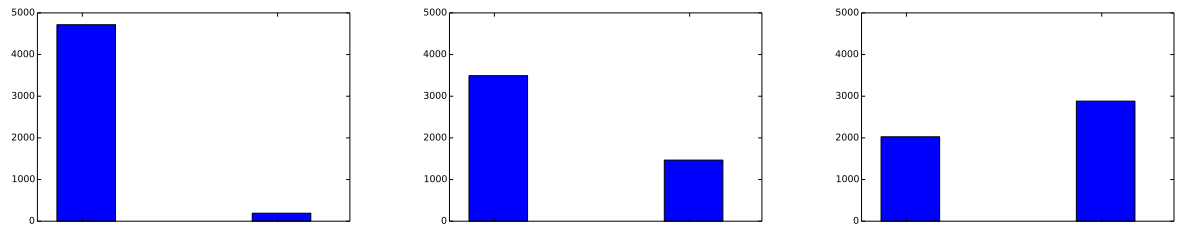

Figure 3: Distribution of clusters for papers that nowadays would be posted to the galaxy subcategory. From left to right: CASB clusters, kmeans clusters, collaborative clusters.

To create our ground truth, we used a Naive Bayes classifier trained on papers appearing in the arXiv in late 2013 and early 2014, which were manually reclassified by the arXiv moderators. We then ran this Naive Bayes classifier on the papers in our 2009-2010 dataset. Note that, although the Naive Bayes classifier is able to automatically classify papers as to whether they belong in astro-ph.GA or astro-ph.CO with high accuracy, this classifier required hand-curated training labels from the arXiv moderators, in the form of correct classifications of a large number of papers from 2013 and 2014. In contrast, in this evaluation, CASB and the benchmark methods do not have access to this training data, and instead must make a determination based only on what was available in 2009-2010.

[pf: Let's could just plot % correctly classified for each method, and explain what that means. There are 2 clusters, and so there are two ways of assigning astro-ph.CO and astro-ph.GA to each. For each one, you can compute % correctly classified by just calculating the number of papers that are in the correct cluster. Then, take the maximum across the two ways of assigning names to clusters.]

Figure 3 shows that 96% of the misplaced papers belong to the same CASB cluster, whereas only 70% of the misplaced papers belong to the same kmeans cluster and only 59% belong to the same collaborative cluster. Thus, we see that in this dataset, using both content and collaborative interaction data greatly improves performance. We also see that CASB is able to reproduce the ground truth with high accuracy using only the 2009-2010 data, reproducing what Naive Bayes required a large amount of human-provided training labels that were not available until 2013-2014.

5 Discussion

Here we have presented a content-augmented stochastic blockmodel, a novel method of clustering items that takes both item similarity and user interaction data into account. The method hinges on the idea that users exist in communities and items exist in clusters, such that the interest of a user in an item is completely determined by the community membership of the user and the cluster membership of the item. We constructed a Gibbs sampler for inference, and tested the model in a real-life scenario with data taken from arXiv.org and the annual INFORMS meeting.

Testing on real-life data showed favourable results for the CASB over other baseline clusters. Though model fit was not perfect, CASB clusters provide a stable middle ground between purely collaborative SB clusters and purely content-based kmeans clusters. On the other hand, the CASB clusters provided strong user interest partitions, suggesting their use could be fruitful in building an item browser. Finally, we found that the CASB clusters were highly effective at identifying would-be misplaced papers on the arXiv, placing 96% of the misplaced papers in the same cluster, whereas kmeans clusters only placed 70% in the same cluster and collaborative clusters only 59%.

This work is a first step at combining stochastic blockmodels with item content. The feature vectors we considered were favourable due to their simplicity, but future work could possibly extend this model to consider item content as a topic model, incorporating latent Dirichlet allocation or one of its variants. Further extensions to the model could be made, such as mixed-membership stochastic blockmodels that allow users to belong to more than one community. This model would be a closer approximation to the real world since, for example, researchers on arXiv.org often fall into several different communities with varying degrees of interest.

References

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.

arXiv.org (2014). arxiv.org e-print archive. `http://arxiv.org`. Accessed: 2014-06-30.

Balabanović, M. and Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*.

Basilico, J. and Hofmann, T. (2004). Unifying collaborative and content-based filtering. In *Twenty-first International Conference on Machine learning - ICML '04*, page 9, New York, New York, USA. ACM Press.

Claypool, M., Gokhale, A., and Miranda, T. (1999). Combining content-based and collaborative filters in an online newspaper. *Proceedings of ACM SIGIR workshop on recommender systems.*, 60.

Ginsparg, P. (2014). personal communication.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

Melville, P., Mooney, R. J., and Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI*, pages 187–192.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Pazzani, M. and Billsus, D. (2007). *Content-based recommendation systems*, pages 325–241. Springer.

Salter, J. and Antonopoulos, N. (2006). CinemaScreen recommender agent: combining collaborative and content-based filtering. *Intelligent Systems, IEEE*, 21(1):35–41.

Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009(4).

Wang, C. and Blei, D. (2011). Collaborative Topic Modeling for Recommending Scientific Articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM.