Proceedings of the 26th AAAI Conference on Artificial Intelligence, Toronto, Canada, July 2012, pp. 879--885.

Clustering Documents Along Multiple Dimensions

Sajib Dasgupta
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120 USA
sdasgup@us.ibm.com

Richard M. Golden
School of Behavioral and Brain Sciences
University of Texas at Dallas
Richardson, TX 75080 USA
golden@utdallas.edu

Vincent Ng
Human Language Technology Institute
University of Texas at Dallas
Richardson, TX 75080 USA
vince@hlt.utdallas.edu

Abstract

Traditional clustering algorithms are designed to search for a single clustering solution despite the fact that multiple alternative clustering solutions might exist for a particular dataset. For example, a set of news articles might be clustered by topic or by the author's gender or age. Similarly, book reviews might be clustered by sentiment or comprehensiveness. In this paper, we address the problem of identifying alternative clustering solutions by developing a Probabilistic Multi-Clustering (PMC) model that discovers multiple, maximally different clusterings of a data sample. Empirical results on six datasets representative of real-world applications show that our PMC model exhibits superior performance to comparable multi-clustering algorithms.

Introduction

Clustering is an established statistical methodology arising in areas such as unsupervised learning, data compression, and exploratory data analysis. Traditional work on clustering has largely focused on generating a single clustering solution of a data sample based upon a particular clustering objective. However, many real-world datasets can be naturally clustered along multiple *dimensions*. For example, a collection of book reviews might be clustered along the dimensions of sentiment (e.g., positive or negative) or genre. Similarly, political blog postings can be clustered by topic, the author's stance (e.g., support or oppose), or her political affiliation (e.g., Democrat or Republican). Traditional text clustering algorithms have often focused on producing a *topic-based* clustering of a dataset, thus failing to satisfy the user's other information needs.

Consequently, researchers started to address the problem of how a clustering algorithm can produce a clustering along the user-desired dimension. Common approaches to this problem include (1) manually identifying the features relevant to the desired dimension (Liu et al. 2004), or (2) *learning* a similarity metric from *side* information (Xing et al. 2002) such as user-defined constraints on which pairs of data items must or must not appear in the same cluster in the user-desired clustering (e.g., Wagstaff et al. (2001), Bilenko et al. (2004)). More recent work has focused on *active* clustering algorithms, where user feedback is incorporated *during* each clustering iteration, specifically by having the user (1) incrementally construct a set of features relevant to the desired dimension in an interactive fashion (e.g., Bekkerman et al. (2007), Raghavan and Allan (2007)), or (2) correct the mistakes made by the algorithm via specifying whether two existing clusters should be *merged* or *split* (e.g., Balcan and Blum (2008)).

While these algorithms can produce a clustering that satisfies a user's interest, they remain somewhat unsatisfactory for several reasons. First, they are knowledge intensive, requiring a lot of human feedback or labeled instances *prior to* or *during* the clustering process. Second, they can only produce a single clustering of a dataset. Hence, if a user wants to cluster a data sample along multiple dimensions, she has to apply the clustering algorithm multiple times, each time requiring new user feedback to produce one of her desired clusterings. Motivated by these inadequacies, we pursue a challenging alternative: we aim to design a fully unsupervised *multi-clustering* model that can generate a set of clusterings of a data sample without relying on any human knowledge for fine-tuning the similarity function or selecting the relevant features, such that each clustering is (1) qualitatively strong in terms of basic qualitative criteria typically used to evaluate the *structure* of a clustering and (2) dissimilar to other clusterings in the set.

To this end, we introduce in this paper an end-to-end probabilistic multi-clustering framework, which we will refer to as Probabilistic Multi-Clustering (PMC). Our PMC model has several appealing characteristics. First, it is developed within a probabilistic framework. In addition to supporting Bayesian decision rules (e.g., minimum probability of error rules) for the assignment of data points to clusters, a probabilistic framework provides an explicit characterization of the assumed multi-clustering environment, which may be used for selecting the most appropriate multi-clustering objective function for a given statistical environment. Second, it assumes *a single feature space*, thus obviating the need for manually identifying relevant features for each dimension of interest. Third, while we focus on the application of PMC to text data in this paper, the underlying framework is general enough for PMC to be applicable to data in other domains. Empirical results on six text datasets demonstrate the superiority of PMC to existing multi-clustering algorithms.

The rest of the paper is structured as follows. We discuss related work, formulate the multi-clustering problem, describe a standard mixture of Gaussians clustering model, extend it to create our multi-clustering model, and present experimental results and our conclusions.

Related Work

To date, there have only been a handful of attempts to tackle the multi-clustering problem. Broadly, these attempts fall into two major categories. In *semi-supervised* approaches, one of the clusterings is provided (by the human) as input, and the goal is to produce the other clustering that is distinctively different from the given one. For instance, Gondek and Hofmann's (2004) approach learns a non-redundant clustering that maximizes the conditional mutual information $I(C;Y|Z)$, where C, Y and Z denote the clustering to be learned, the relevant features and the known clustering, respectively. On the other hand, Davidson and Qi's (2007) approach first learns a distance metric D_C from the original clustering C, and then reverses the transformation of D_C using the Moore-Penrose pseudo-inverse to get the new distance metric D'_C, which is used to produce a distinctively different clustering.

In contrast, our PMC model does not rely on labeled instances or human feedback. Hence, the second category of existing multi-clustering algorithms, *unsupervised* approaches, is more closely related to our work. One of the most well-known unsupervised multi-clustering approach is Caruana et al.'s (2006) meta clustering algorithm, which produces multiple clusterings of a data sample by running k-means multiple times, each time with a random selection of seeds and a random weighting of features. The goal is to present each local minimum of k-means as a possible clustering. Though conceptually simple, meta clustering fails to ensure that the resulting clusterings are dissimilar to each other, a property desirable of a multi-clustering solution. In fact, k-means tends to produce similar clusterings regardless of the number of times it is run.

In comparison to meta clustering, Jain, Meka, and Dhillon's (2008) approach is more sophisticated, as it tries to learn two clusterings in a "decorrelated" k-means framework. Specifically, its joint optimization model ensures that the centroids of the two clusterings are dissimilar while achieving typical k-means objectives. Note that Jain, Meka, and Dhillon use this framework to produce only two clusterings per dataset, since the complexity of their objective function grows with the number of clusterings to be produced.

More recently, Dasgupta and Ng (2010) have shown that the principal eigenvectors of the graph Laplacian reveal the important clustering structures of a data sample, and the eigenvectors, being orthogonal to each other, can be discretized separately to produce a set of distinct 2-way clusterings. Despite its simplicity, their approach is grounded on a particular form of graph-theoretic objective, and it is not clear how to extend it to optimize other forms of objectives or produce multi-way clusterings. In contrast, ours is based on probabilistic modeling, and extending it to produce soft clusterings and multi-way clusterings is natural.

Problem Formulation

In this section, we define the multi-clustering problem formally. Let $X_n \equiv \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a *data sample*, where *data point* $\mathbf{x}_i \in \Re^d$ for $i = 1, \ldots, n$. A *multi-clustering solution* is a set $\mathbf{K} \equiv \{K^{(1)}, \ldots, K^{(z)}\}$ of z clusterings, where each clustering $K^{(m)}$ consists of a finite set of k_m *clusters* $\{C_1^{(m)}, \ldots, C_{k_m}^{(m)}\}$. In essence, the superscript identifies a particular clustering in the set $\mathbf{K}$, and the subscript identifies a particular cluster within a clustering. The number of clusterings to be produced, z, and the number of clusters within each clustering, k_m, are specified by the user. We begin by specifying three constraints for a preference relation on the space of possible multi-clustering solutions.

- *Quality:* Let $\mathbf{K}_1$ and $\mathbf{K}_2$ be multi-clustering solutions. Let Q be a *clustering quality preference function* which is defined such that: $\mathbf{K}_1$ provides a better quality fit to the data than $\mathbf{K}_2$ if and only if $Q(\mathbf{K}_1) \geq Q(\mathbf{K}_2)$. For example, Q might be a log-likelihood function as in the Expectation-Maximization (EM) algorithm (Dempster, Laird, and Rubin 1977) or a least square distortion function as in the K-means algorithm (MacQueen 1967).
- *Distinctivity:* Let $\mathbf{K}_1$ and $\mathbf{K}_2$ be multi-clustering solutions. Let ϕ_K be a *clustering similarity preference function* which is defined such that: The clusterings in $\mathbf{K}_1$ are more dissimilar than the clusterings in $\mathbf{K}_2$ if and only if $\phi_K(\mathbf{K}_1) \leq \phi_K(\mathbf{K}_2)$. We will discuss how ϕ_K can be defined shortly.
- *Parsimony:* Our use of the term "parsimony" is equivalent to the usage of regularization in the machine learning literature. Specifically, we define parsimony constraints that prefer models that are resistant to overfitting.

Diagonal Gaussian Mixture Model

To generate a single clustering of a dataset probabilistically, the standard approach is to employ a Gaussian mixture model (GMM). In this section, we will review how a single clustering of a dataset can be generated using the diagonal Gaussian mixture model (DGMM) and show how to extend this model to find a multi-clustering solution of a dataset in the next section.

Definition. A Gaussian mixture model is a weighted sum of k component Gaussian densities:

$$p(\mathbf{x}|\theta) = \sum_{j=1}^{k} p(C_j|\gamma) p(\mathbf{x}|C_j, \mu_j, \sigma_j),$$

where $\mathbf{x}$ is a d-dimensional continuous-valued feature vector; $p(C_j|\gamma), j = 1, \ldots, k$ are the mixture weights; and $p(\mathbf{x}|C_j, \mu_j, \sigma_j), j = 1, \ldots, k$ are the component Gaussian densities. Each component density is a Gaussian function of the form

$$p(\mathbf{x}|C_j, \mu_j, \sigma_j)$$

$$= \frac{exp[-(1/2)(\mathbf{x}-\mu_j)^T[\mathbf{D}_{\sigma,j}]^{-1}(\mathbf{x}-\mu_j)]}{(2\pi)^{d/2}|\mathbf{D}_{\sigma,j}|^{1/2}},$$

with mean vector μ_j and covariance matrix $\mathbf{D}_{\sigma,j}$. Since we employ a DGMM, $\mathbf{D}_{\sigma,j}$ is a diagonal matrix whose on-diagonal elements are the elements of the d-dimensional vector σ_j^2. The mixture weights are defined in terms of γ, a k-dimensional vector encoding the relevance or importance of each component. Specifically,

$$p(C_j|\gamma) = \frac{\exp[\gamma_j]}{\sum_{u=1}^{k}\exp[\gamma_u]}.$$

Given this definition, it should be easy to see that the DGMM defines a two-stage generative story. First, a cluster C_j is generated with probability $p(C_j|\gamma_j^*)$ for some γ_j^*. Given C_j, a data point $\mathbf{x}$ is generated with probability $p(\mathbf{x}|C_j,\mu_j^*,\sigma_j^*)$ for some μ_j^* and σ_j^*. The DGMM parameters are collectively defined by $\theta \equiv [\theta_1,\ldots,\theta_k] \in \Re^{k(2d+1)}$, where $\theta_j \equiv (\mu_j,\sigma_j,\gamma_j)$ for $j = 1,\ldots,k$. In other words, θ is defined by the mean vectors, the covariance matrix, and the relevance from all component densities.

Parameter estimation. θ can be estimated using maximum likelihood estimation. Let $\mathbf{X}_n \equiv [\mathbf{x}_1,\ldots,\mathbf{x}_n]$ be a data sample in which the n data points are a realization of n independent and identically distributed real-valued random vectors. The likelihood of $\mathbf{X}_n$, $L(\mathbf{X}_n|\theta)$, is given by:

$$L(\mathbf{X}_n|\theta) = \prod_{i=1}^{n} p(\mathbf{x}_i|\theta).$$

By definition, a *maximum likelihood estimate* $\hat{\theta}$ is:

$$\hat{\theta} \equiv \underset{\theta\in\Theta}{\operatorname{argmax}}\, L(\mathbf{X}_n|\theta).$$

$\hat{\theta}$ can be computed using EM, Generalized EM (GEM), or gradient descent.

Clustering. After maximum likelihood estimation, we can use the resulting fitted probability model to induce a clustering on a set of data points. Specifically, the probability that a given data point $\mathbf{x}_i$ is assigned to the jth cluster, C_j, is given by:

$$p(C_j|\mathbf{x}_i,\hat{\theta}_j) = \frac{p(\mathbf{x}_i|C_j,\hat{\theta}_j)p(C_j|\gamma_j)}{p(\mathbf{x}_i)}.$$

To induce a hard clustering, we assign $\mathbf{x}_i$ to the cluster C_j such that the probability $p(C_j|\hat{\theta}_j,\mathbf{x}_i)$ is the largest among all $j = 1,\ldots,k$.

Multi-Clustering DGMM Theory

We now propose a new theory of multi-clustering using DGMMs based on the three multi-clustering criteria described previously, namely quality, distinctivity, and parsimony.

For convenience, let us first introduce some notation. Let $\mathbf{X}_n \equiv [\mathbf{x}_1,\ldots,\mathbf{x}_n]$ be a data sample, as defined above. Recall from the previous section that to induce a single clustering we employ a DGMM. Hence, to induce z clusterings of $\mathbf{X}_n$ we employ z DGMMs. Since each DGMM generates $\mathbf{X}_n$ independently of the other DGMMs, it is convenient to make z copies of each data point $\mathbf{x}_i$, such that its mth copy, $\mathbf{x}_i^{(m)}$, is associated with the mth clustering ($m = 1,\ldots,z$). We denote the set of n points associated with the mth clustering, namely, $\mathbf{x}_1^{(m)},\ldots,\mathbf{x}_n^{(m)}$, as $\mathbf{X}_n^{(m)}$. Moreover, we denote the likelihood of the observed data $\mathbf{X}_n^{(m)}$ given the mth DGMM as $L(\mathbf{X}_n^{(m)}|\theta^{(m)})$, $m = 1,\ldots,z$.

The likelihood of the observed data $\mathbf{X}_n^{(1)},\ldots,\mathbf{X}_n^{(m)}$ given all z DGMMs can therefore be computed by the formula:

$$L(\mathbf{X}_n|\theta) \equiv \prod_{m=1}^{z} L(\mathbf{X}_n^{(m)}|\theta^{(m)}), \qquad (1)$$

where $\theta \equiv [\theta^{(1)},\ldots,\theta^{(z)}] \in \Theta \subseteq \Re^k$, and $k = (2d+1)\sum_{m=1}^{z} k_m$. Θ is called the *parameter space*, and θ defines what we call a *multi-clustering DGMM* (MDGMM).

Finding an MDGMM that maximizes $L(\mathbf{X}_n|\theta)$ will likely produce a multi-clustering solution that has a good quality fit to the given data sample. However, it does not guarantee that the remaining two multi-clustering criteria are satisfied: it is not regularized, as is clear from the objective function; and it does not guarantee distinctivity, because the z clusterings were generated independently by z DGMMs. Consequently, we define a prior p_θ for the MDGMM that incorporates both *distinctivity constraints*, which favor multi-clustering solutions comprising dissimilar clusterings, and *parsimony constraints*, which regularize the parameter space.

Distinctivity constraints. One way to guarantee distinctivity is to ensure that the cluster locations of a pair of clusterings in the multi-clustering solution are dissimilar. To implement this idea, we need to perform two steps: (1) we define a cluster location similarity function for measuring the similarity of two cluster locations, and (2) we minimize this function to penalize multi-clustering solutions containing multiple clusterings with similar cluster location patterns.

Let us begin with step 1. A natural way to define $\phi_{\mu,j}^{(m)}$, the soft clustering cluster-location similarity preference function, is as follows:

$$\phi_{\mu,j}^{(m)} = \left(\frac{1}{m-1}\right)\sum_{u=1}^{m-1}\frac{1}{k_u}\sum_{q=1}^{k_u}\left(\mu_j^m \cdot \mu_q^u\right)^2$$

for $m = 2,3,\ldots,z$ and $\phi_{\mu,j}^{(1)} = 0$. Informally, in $\phi_{\mu,j}^{(m)}$, we first compute one similarity value between the location of the jth cluster in the mth clustering and the location of each cluster in each of the $(m-1)$ clusterings induced so far, and then take the average of these similarity values. It should not be difficult to see that larger values of $\phi_{\mu,j}^{(m)}$ indicate that the cluster locations between clusterings are more similar. We will defer the discussion of step 2 (function minimization) until after we define the parsimony constraints.

Parsimony constraints. To improve finite-sample generalization performance, we perform L2-regularization on the parameter space, which can be encoded as soft parsimony constraints. Since the DGMM that generates the mth clustering has three sets of parameters, $\mu_j^{(m)}$, $\sigma_j^{(m)}$, and $\gamma^{(m)}$, for $j = 1, \ldots, k_m$, we define a regularization function for each set of parameters. Specifically,

$$\eta_{\mu,j}^{(m)} = \|\mu_j^{(m)}\|^2, \eta_{\sigma,j}^{(m)} = \|\sigma_j^{(m)}\|^2, \eta_\gamma^{(m)} = \|\gamma^{(m)}\|^2$$

Minimizing these functions is equivalent to favoring sparse multi-clustering solutions.

Combining the constraints. These preference functions that enable us to incorporate distinctivity and parsimony are not meant to be minimized independently from each other. Rather, we want to combine them as a prior, specifically by defining the *soft clustering solution preference function* $V(\theta)$ such that:

$$V(\theta) \equiv \sum_{m=1}^{z} \sum_{j=1}^{k_m} V_j^{(m)}$$

where

$$V_j^{(m)} = \lambda_\mu^\phi \phi_{\mu,j}^{(m)} + \lambda_\mu^\eta \eta_{\mu,j}^{(m)} + \lambda_\sigma^\eta \eta_{\sigma,j}^{(m)} + \lambda_\gamma^\eta \eta_\gamma^{(m)},$$

and λ_μ^ϕ , λ_μ^η, λ_σ^η and λ_γ^η are non-negative real numbers that are to be determined based on the user's prior knowledge of the importance of each constraint. Define the Bayesian parameter prior p_θ as a monotonically decreasing function of $V(\theta)$ such that:

$$p_\theta(\theta) \equiv \frac{1}{Z} \exp\left(\frac{-V(\theta)}{2\sigma_p^2}\right),$$

where σ_p controls the intensity of the prior. The lower the value of σ_p the higher is the effect of the distinctivity and parsimony constraints. The normalization constant Z exists provided the parameter space Θ is compact.

Modifying the objective function. Next, we incorporate the prior p_θ into the objective function. The use of p_θ enables us to seek a MAP (maximum a posteriori) estimate, $\hat{\theta}$, rather than a maximum likelihood estimate. By definition, a MAP estimate $\hat{\theta}$ is:

$$\begin{aligned} \hat{\theta} &\equiv \underset{\theta\in\Theta}{\operatorname{argmax}}\, p(\theta|\mathbf{X}_n) \\ &= \underset{\theta\in\Theta}{\operatorname{argmax}}\, p(\mathbf{X}_n|\theta)p_\theta(\theta) \\ &= \underset{\theta\in\Theta}{\operatorname{argmax}} \prod_{m=1}^{z} L(\mathbf{X}_n|\theta^{(m)})p_\theta(\theta). \end{aligned}$$

As usual, we maximize the logarithm of this objective function, which is

$$\log \prod_{m=1}^{z} L(\mathbf{X}_n|\theta^{(m)}) + \log p_\theta(\theta). \qquad (2)$$

MAP estimation strategies. Next, we discuss our strategy for estimating the model parameters given the aforementioned MAP objective. We employ the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm for parameter estimation, which has a superlinear convergence rate (Luenberger 1984). Note that parameter estimation is especially challenging because the MAP objective function tends to be plagued by multiple local and global maxima as well as saddle points. To address this challenge, we developed a *5-step* multi-stage estimation algorithm, as shown below. The idea is to generate a series of progressively more complicated estimation problems using the results of the previously solved problem as an initial guess for the next more complicated problem.

- **Step 1:** Let $m = 0$. Define an initial guess $\hat{\theta}^{(1)}$ for clustering 1.
- **Step 2:** Let $m = m + 1$.
- **Step 3:** Use the BFGS algorithm to estimate $\theta^{(m)}$ using $\hat{\theta}^{(m-1)}, \ldots, \hat{\theta}^{(1)}$ as an initial guess by seeking a $\hat{\theta}^{(m)}$ that maximizes:

  $$p(\theta^{(m)}|\mathbf{X}_n, \hat{\theta}^{(m-1)}, \ldots, \hat{\theta}^{(1)}) = \frac{p(\mathbf{X}_n|\theta^{(m)}, \hat{\theta}^{(m-1)}, \ldots, \hat{\theta}^{(1)})p(\theta^{(m)}|\hat{\theta}^{(m-1)}, \ldots, \hat{\theta}^{(1)})}{p(\mathbf{X}_n)},$$

  or equivalently, use the objective function in (2) to maximize $p(\mathbf{X}_n|\theta^{(m)})p(\theta^{(m)}, \hat{\theta}^{(m-1)}, \ldots, \hat{\theta}^{(1)})$ with respect to $\theta^{(m)}$, holding $\hat{\theta}^{(m-1)}, \ldots, \hat{\theta}^{(1)}$ constant.
- **Step 4:** Go to Step 2 until $m = z$; Else go to Step 5.
- **Step 5:** Use the BFGS algorithm, initial guess $\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(z)}$, and the objective function in (2) to maximize:

  $$p(\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(z)}|\mathbf{X}_n)$$

  with respect to $\theta^{(1)}, .., \theta^{(z)}$.

Evaluation

Experimental Setup

Datasets. We employ six evaluation datasets that cover a variety of clustering dimensions.

Two Newsgroups (TNG) consists of all the documents from two sections of 20 Newsgroups, `talks.politics` and `sci.crypt`.

Blitzer, Dredze, and Pereira's (2007) book (BOOK) and DVD datasets each contains 1000 positive and 1000 negative customer reviews of books or movies, and can therefore be used to evaluate our model's ability to cluster by *sentiment*. Since we desire that each dataset possesses at least two clustering dimensions, we also manually annotate each review with a *subjectivity* label that indicates whether it is "mostly subjective" (where the reviewer mainly expresses her sentiment) or "mostly objective" (where the reviewer focuses on describing the content of the book or the movie). Details of the annotation process are described later in this subsection.

The MIX dataset is a 4000-document dataset consisting of the 2000 BOOK reviews and the 2000 DVD reviews, as described above. We can therefore cluster these reviews by *topic* (i.e., book or DVD), *sentiment* or *subjectivity*.

Schler et al.'s (2006) MAN dataset contains 19,320 blog posts. We randomly selected 1000 blog postings, half of which were written by males and half by females. We can therefore cluster these blog posts by the author's *gender*. Since the author's *age* information is also available in each

TNG	Topic1
BOOK	Sentiment, Subjectivity
DVD	Sentiment, Subjectivity
MIX	Topic2, Sentiment, Subjectivity
MAN	Gender, Age
POL	Political Affiliation, Policy

Table 1: Clustering dimensions for the six datasets.

blog post, we can also cluster them by age. To do so, we automatically generate a 2-way partitioning of the documents by imposing an age threshold of 25. Specifically, the 932 documents written by bloggers aged below 25 are marked as *young*, and the remaining 1068 are marked as *old*.

Our own POL dataset consists of 2000 political articles written by columnists, 1000 of whom identified themselves as *Republicans* and the remaining 1000 identified themselves as *Democrats*.[1] Hence, we can cluster these articles by the author's *political affiliation*. We also create a second clustering dimension by annotating each article as either *foreign* or *domestic*, depending on the policy that the article discusses. For example, the policy on the Iraq war is foreign, whereas the policy on regulating the job market is domestic.

Table 1 shows the dimensions along which the documents are annotated. Each of the eight distinct dimensions yields a 2-way partitioning of the documents: (1) Topic1 (science/politics); (2) Sentiment (positive/negative); (3) Subjectivity (subjective/objective); (4) Topic2 (book/DVD); (5) Gender (man/woman); (6) Age (young/old); (7) Political affiliation (Democrat/Republican); and (8) Policy (domestic/foreign).

Human annotation. As noted above, we need to annotate the BOOK, DVD, and MIX datasets with respect to *Subjectivity* and POL with respect to *Policy*.[2] We had two computer science graduate students independently annotate the documents. For POL, we asked them to use commonsense knowledge to annotate each document with respect to the policy that the article discusses. If both foreign and domestic policies are discussed in the text, we asked them to assign the label based on the one that is discussed more frequently. On the other hand, given a BOOK or DVD review, we asked them to first label each of its sentences as subjective or objective; if a sentence contains both subjective and objective materials, its label should reflect the type of material that appears more frequently. The review is then labeled as subjective (objective) if more than half of its sentences are labeled as subjective (objective).

Document preprocessing. To preprocess a document, we follow the standard procedure. We tokenize and downcase it, remove stopwords, and represent it as a vector of unstemmed unigrams. Also, following common tradition in high dimensional data clustering (Huber 1985; Dasgupta 1999), we project the data matrix into a lower dimensional subspace, specifically by applying Singular Value Decomposition to the matrix and employing the top 25 singular vectors as our projected subspace.

Evaluation metrics. We employ two evaluation metrics. First, we report results for each dataset in terms of accuracy, which is the fraction of documents for which the label assigned by our system is the same as the gold-standard label. Second, following Kamvar, Klein, and Manning (2003), we evaluate the clusters produced by our approach against the gold-standard clusters using the Adjusted Rand Index (ARI) (Hubert and Arabie 1985). ARI is the adjusted-for-chance form of the Rand Index (Rand 1971), which computes the pairwise accuracy given two partitions. ARI ranges from -1 to 1; better clusterings have higher values.

Performance of Three Baseline Algorithms

As baselines, we employ meta clustering (Caruana et al. 2006), GMM-EM, and Davidson and Qi's (2007) algorithm.[3]

Meta clustering produces multiple clusterings of a data sample by running k-means multiple times, each time with a random selection of seeds and a random weighting of features (Caruana et al. 2006). We ran it 100 times and reported in row 1 of Tables 2 and 3 the *best* accuracy and ARI obtained for each dimension of each dataset. Although the best results are reported, it performs poorly in general. The poor performance can be attributed in part to the fact that k-means is generally a weaker clustering algorithm than its more recently developed counterparts.

GMM-EM is somewhat similar in spirit to meta clustering: we can use it to produce multiple clusterings of a data sample by running it multiple times, each time with a random parameter initialization. We employ EM for parameter estimation in GMM. For each dataset, we (1) run the system z times to produce z different proposal clusterings; (2) find the one-to-one mapping between the z proposal clusterings and the gold standard clusterings that yields the highest accuracy; and (3) report the accuracy and ARI of a proposal clustering against the mapped gold standard clustering. In all experiments, we set z to 5.[4] The accuracy and ARI results, averaged over three independent runs, are reported in row 2 of Tables 2 and 3. As can be seen, GMM-EM performs well only along the *Subjectivity* dimension for all of the sentiment datasets (i.e., BOOK, DVD and MIX). Its relatively poor performance can be attributed to the fact that the clusterings it produces for each dataset are similar to each other despite being supplied with different initializations.

Davidson and Qi's (2007) algorithm has a somewhat different goal than PMC: it is intended to be used as a semi-

[1] These articles were chosen randomly among those written in 2006 from `http://www.commondreams.org/archives`.

[2] Note that the subjectivity labels for MIX can be derived from BOOK and DVD.

[3] The experimental results for meta clustering and Davidson and Qi's system were obtained using the publicly available implementations that we downloaded from `www.cs.cornell.edu/~nhnguyen/metaclustering.htm` and `www.cs.ucdavis.edu/~davidson/constrained-clustering/CAREER/CAREER.html` respectively.

[4] This choice of z is motivated by one observation: since each of our datasets can be clustered along two or three dimensions, we hypothesize that a good multi-clustering model should be able to generate most, if not all, of the clusterings when z is set to 5.

	TNG	BOOK		DVD		MIX			MAN		POL	
System	Topic	Sent.	Subj.	Sent.	Subj.	Topic	Sent.	Subj.	Gend.	Age	Affil.	Policy
Meta clustering	76.2	50.8	51.2	53.9	71.0	50.2	50.2	58.6	51.2	53.6	59.4	58.8
GMM-EM	**79.1**	50.3	**67.4**	53.3	**72.2**	50.7	52.1	**69.5**	51.8	52.8	55.1	57.1
Davidson & Qi	—	50.9	55.7	51.2	59.6	50.5	51.2	57.8	50.5	50.9	50.5	51.2
PMC	72.3	**59.1**	61.5	**57.1**	61.7	**64.8**	**60.5**	63.5	**63.0**	**56.5**	**64.6**	**62.2**

Table 2: Results in terms of accuracy. The best result for each dimension is boldfaced.

	TNG	BOOK		DVD		MIX			MAN		POL	
System	Topic	Sent.	Subj.	Sent.	Subj.	Topic	Sent.	Subj.	Gend.	Age	Affil.	Policy
Meta clustering	0.275	0.001	0.001	0.006	0.155	0.001	0.001	0.021	0.001	0.003	0.035	0.023
GMM-EM	**0.304**	0.001	**0.117**	0.004	**0.187**	0.001	0.002	**0.144**	0.001	0.001	0.010	0.008
Davidson & Qi	—	0.001	0.006	0.001	0.020	0.001	0.001	0.020	0.001	0.001	0.001	0.001
PMC	0.237	**0.032**	0.050	**0.020**	0.049	**0.087**	**0.044**	0.071	**0.067**	**0.015**	**0.086**	**0.058**

Table 3: Results in terms of Adjusted Rand Index (ARI). The best result for each dimension is boldfaced.

supervised clustering algorithm. As mentioned in Related Work, given a dataset with two clusterings, the algorithm assumes that one of these clusterings is supplied (by a human) as input and aims to produce the other clustering. Following its intended use, for each dataset with clustering dimensions i and j, we feed it with the gold-standard clustering for dimension i and measure the quality of the clustering it proposes against the gold-standard clustering for dimension j. The same experiment is repeated with the roles of i and j switched. As we can see from row 3 of Tables 2 and 3, despite its access to gold-standard clusterings, it does not perform any better than the other baselines.

Performance of the PMC Model

To obtain the results of our PMC model, we follow the same evaluation procedure as conducted for GMM-EM. In these investigations, we have chosen to focus on understanding the role of only a subset of the distinctivity and parsimony constraints, so we assume that (1) λ^{η}_{σ} and λ^{η}_{γ} are zero, and (2) λ^{ϕ}_{μ} and λ^{η}_{μ} are one. We set $\sigma_j^{(m)}$ to 0.1 and $\gamma_j^{(m)}$ to 0.5, keeping them constant during the learning process as the optimization is highly sensitive to slight perturbations to these parameters. Moreover, we set σ_p to 0.01. As in GMM-EM, we set z to 5. As BFGS optimization is sensitive to parameter initialization, we repeat the experiments three times for each dataset and report the average results.

Results of PMC are shown in row 4 of Tables 2 and 3. In comparison to the best baseline for each clustering dimension, PMC achieved the best result for 8 of the 12 dimensions. More importantly, it achieved stable performance across different dimensions of a dataset. For example, for the MIX dataset, PMC achieved accuracies of 64.8%, 60.5% and 63.5% for the topic, sentiment and subjectivity dimensions respectively, whereas the baselines obtained $< 60\%$ accuracies on all three dimensions. Similar trends can be observed for the ARI results.

As mentioned above, the primary reason why GMM-EM performed poorly was that the clusterings it produced were similar to each other. On the other hand, we hypothesize that our PMC model does not have this problem, owing in part to the design criterion of distinctivity.

MIX			
$K^{(1)}$	$K^{(2)}$	$K^{(3)}$	$K^{(4)}$
$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$
text	watched	mind	relationship
knowledge	music	novel	lives
death	bought	readers	human
national	episode	uses	wonderful
information	version	reader	women
using	wanted	nature	age
case	show	human	view
$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$
films	readers	young	disappointed
scene	national	features	caught
reason	ask	military	action
ending	facts	disc	novel
script	reader	video	mystery
absolutely	destruction	screen	isn
comedy	christ	classic	killer
Topic2	**Subjectivity**	**Topic2**	**Sentiment**

Table 4: Top features representing the clustering $K^{(1)}, .., K^{(4)}$ for the MIX dataset. Each clustering comprises two clusters, i.e., $\mathbf{C}_1$ and $\mathbf{C}_2$.

To better understand whether PMC indeed produces distinct clusterings, we show in Table 4 the top words representing the clusterings generated by PMC for the MIX dataset. Specifically, PMC discovers four distinct clusterings $K^{(1)}$, . . ., $K^{(4)}$ of the MIX dataset where each clustering consists of two clusters, C_1 and C_2. For each clustering, the top words are selected on the basis of their relative frequency of occurrence in each cluster. The dimension labels in the last row are manually determined by inspecting the top words.

Note that the most important words associated with the clusters of one clustering are quite different from those associated with the clusters of another clustering. A closer examination of the results reveals that the same is true for the remaining datasets with at least two clustering dimensions. This observation illustrates the idea that PMC is not biased towards generating a particular type of clustering but rather its performance is consistent with the design constraints embedded within PMC. These constraints bias PMC to generate distinctive clusterings. Moreover, it is possible that the top words generated for each clustering might be interpreted as PMC's "discovery" of new feature spaces, but additional research is needed to thoroughly explore this idea.

Conclusions

We have presented PMC, a generative probabilistic model for producing multiple clusterings of a data sample, demonstrating how to incorporate as soft constraints three properties highly desired of a multi-clustering solution, namely distinctivity, quality, and parsimony. A probabilistic approach to multi-clustering has a number of important advantages, including (1) Bayesian decision rules for classifying data points into clusters, (2) explicit specifications of probabilistic modeling assumptions, and (3) the opportunity to exploit Bayesian model selection criteria for selecting the most appropriate multi-clustering objective function for a particular statistical environment. We showed that our PMC model achieved more stable results along different dimensions of a variety of text datasets in comparison to three existing multi-clustering algorithms. Considering the fact that multi-clustering is a challenging problem, we believe that the results obtained by PMC thus far are very promising. Equally importantly, we believe that our probabilistic formulation provides a fresh perspective on the multi-clustering problem and sets the stage for further investigation.

Acknowledgments

We thank the three anonymous reviewers for their valuable comments on an earlier draft of this paper.

References

Balcan, M.-F., and Blum, A. 2008. Clustering with interactive feedback. In *Proceedings of the 19th International Conference on Algorithmic Learning Theory*, 316–328.

Bekkerman, R.; Raghavan, H.; Allan, J.; and Eguchi, K. 2007. Interactive clustering of text collections according to a user-specified criterion. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 684–689.

Bilenko, M.; Basu, S.; and Mooney, R. J. 2004. Integrating constraints and machine learning in semi-supervised clustering. In *Proceedings of the 21st International Conference on Machine Learning*, 81–88.

Blitzer, J.; Dredze, M.; and Pereira, F. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, 440–447.

Caruana, R.; Elhawary, M. F.; Nguyen, N.; and Smith, C. 2006. Meta clustering. In *Proceedings of 6th IEEE International Conference on Data Mining*, 107–118.

Dasgupta, S., and Ng, V. 2010. Mining clustering dimensions. In *Proceedings of the 27th International Conference on Machine Learning*, 263–270.

Dasgupta, S. 1999. Learning mixtures of Gaussians. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*.

Davidson, I., and Qi, Z. 2007. Finding alternative clusterings using constraints. In *Proceedings of the 8th IEEE International Conference on Data Mining*, 773–778.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* 39(1):1–38.

Gondek, D., and Hofmann, T. 2004. Non-redundant data clustering. In *Proceedings of the 4th IEEE International Conference on Data Mining*, 75–82.

Huber, P. J. 1985. Projection pursuit. *Annals of Statistics* 13(2):435–475.

Hubert, L., and Arabie, P. 1985. Comparing partitions. *Journal of Classification* 2(1):193–218.

Jain, P.; Meka, R.; and Dhillon, I. S. 2008. Simultaneous unsupervised learning of disparate clusterings. In *Proceedings of SIAM International Conference on Data Mining*, 858–869.

Kamvar, S.; Klein, D.; and Manning, C. 2003. Spectral learning. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 561–566.

Liu, B.; Li, X.; Lee, W. S.; and Yu, P. S. 2004. Text classification by labeling words. In *Proceedings of the 19th National Conference on Artificial Intelligence*, 425–430.

Luenberger, D. G. 1984. *Linear and Nonlinear Programming*. Addison-Wesley, 2nd edition.

MacQueen, J. B. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Symposium on Math, Statistics, and Probability*, 281–297.

Raghavan, H., and Allan, J. 2007. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 79–86.

Rand, W. M. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* 66(336):846–850.

Schler, J.; Koppel, M.; Argamon, S.; and Pennebaker, J. 2006. Effects of age and gender on blogging. In *Proceedings of the 2006 AAAI Symposium on Computational Approaches for Analyzing Weblogs*.

Wagstaff, K.; Cardie, C.; Rogers, S.; and Schrödl, S. 2001. Constrained k-means clustering with background knowledge. In *Proceedings of the 18th International Conference on Machine Learning*, 577–584.

Xing, E. P.; Ng, A. Y.; Jordan, M. I.; and Russell, S. J. 2002. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, 505–512.