Proc. PKDD, Porto, Portugal, 2005

A Probabilistic Clustering-Projection Model for Discrete Data

Shipeng Yu[12], Kai Yu[2], Volker Tresp[2], and Hans-Peter Kriegel[1]

[1] Institute for Computer Science, University of Munich, Germany
[2] Siemens Corporate Technology, Munich, Germany

Abstract. For discrete co-occurrence data like documents and words, calculating optimal projections and clustering are two different but related tasks. The goal of projection is to find a low-dimensional latent space for words, and clustering aims at grouping documents based on their feature representations. In general projection and clustering are studied independently, but they both represent the intrinsic structure of data and should reinforce each other. In this paper we introduce a probabilistic clustering-projection (PCP) model for discrete data, where they are both represented in a unified framework. Clustering is seen to be performed in the projected space, and projection explicitly considers clustering structure. Iterating the two operations turns out to be exactly the variational EM algorithm under Bayesian model inference, and thus is guaranteed to improve the data likelihood. The model is evaluated on two text data sets, both showing very encouraging results.

1 Introduction

Modelling discrete data is a fundamental problem in machine learning, pattern recognition and statistics. The data is usually represented as a large (and normally sparse) matrix, where each entry is an integer and characterizes the relationship between corresponding row and column. For example in document modelling, the "bag-of-words" methods represent each document as a row vector of occurrences of each word, ignoring any internal structure and word order. This is taken as the working example in this paper, but the proposed model is generally applicable to other discrete data.

Data projection and clustering are two important tasks and have been widely applied in data mining and machine learning (e.g., principal component analysis (PCA) and k-means [1]). Projection is also referred as feature mapping that aims to find a new representation of data, which is low-dimensional and physically meaningful. On the other hand, clustering tries to group similar data patterns together, and thus uncovers the structure of data. Traditionally these two methods are studied separately and mainly on continuous data. However in this paper we investigate them on *discrete* data and treat them *jointly*.

Projection on discrete data differs from the case on continuous space, where, for example, the most popular technology PCA tries to find the orthogonal dimensions (or factors) that explains the *covariance* of data dimensions. However,

one cannot make the same orthogonal assumption on the low-dimensional factors of discrete data and put the interests on the covariance anymore. Instead, it is desired to find the *independent* latent factors that explain the *co-occurrence* of dimensions (e.g., words). In text modelling, if we refer the factors as topics, the projection actually represent each document as a data point in a low-dimensional topic space, where a co-occurrence factor actually suggests more or less a cluster of words (i.e., a group of words often occurring together). Intuitively, if the projected topic space is informative enough, it should also be highly indicative to reveal the clustering structure of documents. On the other hand, a truly discovered clustering structure reflects the shared topics within document clusters and the distinguished topics across document clusters, and thus can offer evidence for the projection side. Therefore, it is highly desired to consider the two problems in a unified model.

In this paper a novel probabilistic clustering-projection (PCP) model is proposed, to jointly handle the projection and clustering for discrete data. The projection of words is explicitly formulated with a matrix of model parameters. Document clustering is then incorporated using a mixture model on the projected space, and we model each mixture component as a multinomial over the latent topics. In this sense this is a *clustering model using projected features* for documents if the projection matrix is given, and a *projection model with structured data* for words if the clustering structure is known. A nice property of the model is that we can perform clustering and projection *iteratively*, incorporating new information on one side to the updating of the other. We will show that they are corresponding to a Bayesian variational EM algorithm that improves the data likelihood iteratively.

This paper is organized as follows. The next section reviews related work. Section 3 introduces the PCP model and explicitly points out the clustering and projection effects. In Section 4 we present inference and learning algorithm. Then Section 5 presents experimental results and Section 6 concludes the paper.

2 Related Work

PCA is perhaps the most well-known projection technique, and has its counterpart in information retrieval called latent semantic indexing [4]. For discrete data, an important related work is probabilistic latent semantic indexing (pLSI) [7] which directly models latent topics. PLSI can be treated as a projection model, since each latent topic assigns probabilities to a set of words and thus a document, represented as a bag of words, can be treated as generated from a mixture of multiple topics. However, the model is not built for clustering and, as pointed by Blei et al. [2], it is not a proper generative model, since it treats document IDs as random variables and thus cannot generalize to new documents. Latent Dirichlet allocation (LDA) [2] generalizes pLSI by treating the topic mixture parameters (i.e., a multinomial over topics) as variables drawn from a Dirichlet distribution. This model is a well-defined generative model and performs much better than pLSI, but the clustering effect is still missing. On the other side, doc-

ument clustering has been intensively investigated and the most popular method is probably partition-based algorithms like k-means (see, e.g., [1]). Non-negative matrix factorization (NMF) [11] is another candidate and is shown to obtain good results in [13].

Despite that plenty of work has been done in either clustering or projection, the importance of considering both in a single framework has been noticed only recently, e.g., [6] and [12]. Both works are concerned about document clustering and projection on continuous data, while lacking the probabilistic interpretations to the connections among documents, clusters and factors. Buntine et al. [3] noticed this problem for discrete data and pointed out that the multinomial PCA model (or discrete PCA) takes clustering and projection as two extreme cases. Another closely related work is the so-called two-sided clustering, like [8] and [5], which aims to clustering words and documents simultaneously. In [5] it is implicitly assumed a one-to-one correspondence between the two sides of clusters. [8] is a probabilistic model for discrete data, but it has similar problems as in pLSI and not generalizable to new documents.

3 The PCP Model

We consider a corpus $\mathcal{D}$ containing D documents, with vocabulary $\mathcal{V}$ having V words. Following the notation in [2], each document d is a sequence of N_d words that is denoted by $\mathbf{w}_d = \{w_{d,1}, \ldots, w_{d,N_d}\}$, where $w_{d,n}$ is a variable for the nth word in $\mathbf{w}_d$ and denotes the index of the corresponding word in $\mathcal{V}$.

To simplify explanations, we use "clusters" for components in document clustering structure and "topics" for projected space for words. Let M denote the number of clusters and K the dimensionality of topics. Roman letters d, m, k, n, j are indices for documents, clusters, topics, words in $\mathbf{w}_d$, and words in $\mathcal{V}$. They are up to D, M, K, N_d, V, respectively. Letter i is reserved for temporary index.

3.1 The Probabilistic Model

The PCP model is a generative model for a document corpus. Figure 1 (left) illustrates the sampling process in an informal way. To generate one document d, we first choose a cluster from the M clusters. For the mth cluster, the cluster center is denoted as θ_m and defines a topic mixture over the topic space. Therefore θ_m is a K-dimensional vector and satisfies $\theta_{m,k} \geq 0$, $\sum_{k=1}^{K} \theta_{m,k} = 1$ for all $m = 1, \ldots, M$. The probability of choosing a specific cluster m for document d is denoted as π_m, and $\boldsymbol{\pi} := \{\pi_1, \ldots, \pi_M\}$ satisfies $\pi_m \geq 0$, $\sum_{m=1}^{M} \pi_m = 1$.

When document d chooses cluster m, it defines a document-specific topic mixture θ_d, which is obtained exactly from the cluster center θ_m. Note that everything is discrete and two documents belonging to the same cluster will have the same topic mixtures. Words are then sampled independently given topic mixture θ_d, in the same way as in LDA. Each word $w_{d,n}$ is generated by first choosing a topic $z_{d,n}$ given the topic mixture, and then sampling the word given the projection β. β is the $K \times V$ matrix where $\beta_{k,j}$ specifies the probability

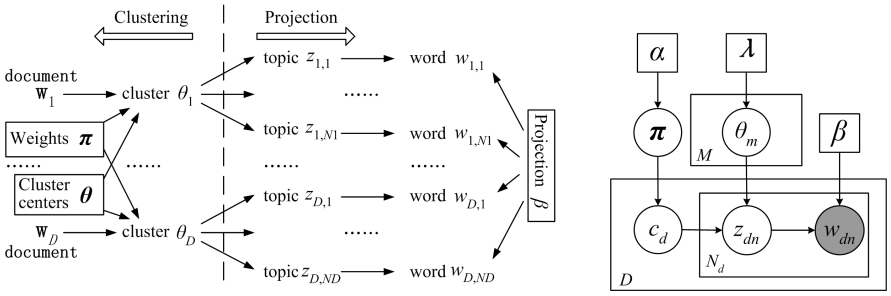


Fig. 1. Informal sampling process (left) and plate model (right) for the PCP model. In the left figure, dark arrows show dependencies between entities and the dashed line separates the clustering and projection effects. In the plate model, rectangle means independent sampling, and hidden variables and model parameters are denoted as circles and squares, respectively. Observed quantities are marked in black.

of generating word j given topic k, $\beta_{k,j} = p(w^j = 1 | z^k = 1)$. Therefore each row $\beta_{k,:}$ defines a multinomial distribution for all words over topic k and satisfies $\beta_{k,j} \geq 0$, $\sum_{j=1}^{V} \beta_{k,j} = 1$.

To complete the model, we put a Dirichlet prior $\mathrm{Dir}(\boldsymbol{\lambda})$ for all the cluster centers $\theta_1, \ldots, \theta_M$, and a symmetric Dirichlet prior $\mathrm{Dir}(\alpha/M, \ldots, \alpha/M)$ for the mixing weights $\boldsymbol{\pi}$. Note that they are sampled only once for the whole corpus.

Finally we obtain the probabilistic model formally illustrated in Figure 1 (right), using standard plate model. c_d takes value $\{1, \ldots, M\}$ and acts as the indicator variable saying which cluster document d takes on out of the M clusters. All the model parameters are $\alpha, \boldsymbol{\lambda}$, β and amount to $1 + M + K \times (V - 1)$. The following procedure describes the sampling process for the whole corpus:

1. Choose model parameter $\alpha, \boldsymbol{\lambda}, \beta$;
2. For the mth cluster, choose $\theta_m \sim \mathrm{Dir}(\boldsymbol{\lambda}), m = 1, \ldots, M$;
3. Choose the mixing weight $\boldsymbol{\pi} \sim \mathrm{Dir}(\alpha/M, \ldots, \alpha/M)$;
4. For each document $\mathbf{w}_d$:
   (a) Choose a cluster m with mixing weights $\boldsymbol{\pi}$, and obtain $\theta_d = \theta_m$;
   (b) For each of the N_d words $w_{d,n}$:
      i. Choose a topic $z_{d,n} \sim \mathrm{Mult}(\theta_d)$;
      ii. Choose a word $w_{d,n} \sim \mathrm{Mult}(\beta_{z_{d,n},:})$.

Denote $\boldsymbol{\theta}$ as the set of M cluster centers $\{\theta_1, \ldots, \theta_M\}$, the likelihood of the corpus $\mathcal{D}$ can be written as

$$\mathcal{L}(\mathcal{D}; \alpha, \boldsymbol{\lambda}, \beta) = \int_{\boldsymbol{\pi}} \int_{\boldsymbol{\theta}} \prod_{d=1}^{D} p(\mathbf{w}_d | \boldsymbol{\theta}, \boldsymbol{\pi}; \beta) dP(\boldsymbol{\theta}; \boldsymbol{\lambda})\, dP(\boldsymbol{\pi}; \alpha), \tag{1}$$

where $p(\boldsymbol{\theta}; \boldsymbol{\lambda}) = \prod_{m=1}^{M} p(\theta_m; \boldsymbol{\lambda})$, and the likelihood of document d is a mixture:

$$p(\mathbf{w}_d | \boldsymbol{\theta}, \boldsymbol{\pi}; \beta) = \sum_{c_d=1}^{M} p(\mathbf{w}_d | \boldsymbol{\theta}, c_d; \beta) p(c_d | \boldsymbol{\pi}). \tag{2}$$

Given mixture component c_d, likelihood term $p(\mathbf{w}_d|\boldsymbol{\theta}, c_d; \beta)$ is then given by

$$p(\mathbf{w}_d|\theta_{c_d}; \beta) = \prod_{n=1}^{N_d} \sum_{z_{d,n}=1}^{K} p(w_{d,n}|z_{d,n}; \beta) p(z_{d,n}|\theta_{c_d}). \tag{3}$$

3.2 PCP as a Clustering Model

As can be seen from (2) and (3), PCP is a clustering model when the projection β is assumed known. The essential terms now are the probabilities of clusters $p(m|\boldsymbol{\pi}) = \pi_m$, probabilistic clustering assignment for documents $p(\mathbf{w}_d|\theta_m; \beta)$, and cluster centers θ_m, for $m = 1, \ldots, M$. Note from (3) that cluster centers θ_m are not modelled directly with words like $p(w|\theta_m)$, but with topics, $p(z|\theta_m)$. This means we are not clustering documents in word space, but in *topic space*. This is analogous to clustering continuous data on the latent space found by PCA [6], and K is exactly the dimensionality of this space. To obtain the probability that document d belongs to cluster m, we project each word into topic space, and then calculate the distance to cluster center θ_m by considering all the words in $\mathbf{w}_d$. This explains (3) from perspective of clustering.

To improve generalization and avoid overfitting, we put priors to θ_m and $\boldsymbol{\pi}$ and treat them as hidden variables, as usually done in mixture modelling. The prior distributions are chosen to be Dirichlet that is *conjugate* to multinomial. This will make model inference and learning much easier (see Section 4).

3.3 PCP as a Projection Model

A projection model aims to learn projection β, mapping words to topics. As can be seen from (3), the topics are not modelled directly with documents $\mathbf{w}_d$, but with cluster centers θ_m. Therefore if clustering structure is already known, PCP will learn β by using the richer information contained in cluster centers, not just individual documents. In this sense, PCP can be explained as a *projection model with structured data* and is very attractive because clustered documents are supposed to contain less noise and coarser granularity. This will make the projection more accurate and faster.

As a projection model, PCP is more general than pLSI because document likelihood (3) is well defined and generalizable to new documents. Although LDA uses similar equation as (3), the topic mixture θ_d is only sampled for current document and no inter-similarity of documents is directly modelled. Documents can only exchange information via the hyperparameter for θ_d's, and thus its effect to β is only implicit. On the contrary, PCP directly models similarity of documents and incorporate all information to learn β.

As discussed in [2], projection β can be smoothed by putting a common prior to all the rows. If only the *maximum a posteriori* (MAP) estimate of β is considered, the effect of smoothing turns out to add a common factor to each entry of β before normalization each row. This is also straightforward in PCP model and we will not discuss it in detail for simplicity. In the experiments we will use this smoothing technique.

4 Inference and Learning

In this section we consider model inference and learning. As seen from Figure 1, for inference we need to calculate the *a posteriori* distribution of latent variables

$$\hat{p}(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z}) := p(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z} | \mathcal{D}, \alpha, \boldsymbol{\lambda}, \beta),$$

including both effects of clustering and projection. Here for simplicity we denote $\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z}$ as groups of $\pi_m, \theta_m, c_d, z_{d,n}$, respectively. This requires to compute (1), where the integral is however analytically infeasible. A straightforward Gibbs sampling method can be derived, but it turns out to be very slow and inapplicable to high dimensional discrete data like text, since for each word we have to sample a latent variable z. Therefore in this section we suggest an efficient *variational method* by introducing variational parameters for latent variables [9]. Then we can maximize the data likelihood by iteratively updating these parameters and obtain a *variational EM* algorithm until convergence. The interesting thing is that this algorithm is equivalent to performing clustering and projection iteratively, which we will discuss in detail.

4.1 Variational EM Algorithm

The idea of variational EM algorithm is to propose a joint distribution $q(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z})$ for latent variables conditioned on some free parameters, and then enforce q to approximate the *a posteriori* distributions of interests by minimizing the KL-divergence $D_{\mathrm{KL}}(q \| \hat{p})$ with respect to those free parameters. We propose a variational distribution q over latent variables as the following

$$q(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z} | \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\psi}, \boldsymbol{\phi}) = q(\boldsymbol{\pi} | \boldsymbol{\eta}) \prod_{m=1}^{M} q(\theta_m | \gamma_m) \prod_{d=1}^{D} q(c_d | \psi_d) \prod_{n=1}^{N_d} q(z_{d,n} | \phi_{d,n}), \quad (4)$$

where $\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\psi}, \boldsymbol{\phi}$ are groups of *variational parameters*, each tailoring the variational *a posteriori* distribution to each latent variable. In particular, $\boldsymbol{\eta}$ specifies an M-dim. Dirichlet for $\boldsymbol{\pi}$, γ_m specifies a K-dim. Dirichlet for distinct θ_m, ψ_d specifies an M-dim. multinomial for indicator c_d of document d, and $\phi_{d,n}$ specifies a K-dim. multinomial over latent topics for word $w_{d,n}$. It turns out that minimization of the KL-divergence is equivalent to maximization of a lower bound of the log likelihood $\ln p(\mathcal{D} | \alpha, \boldsymbol{\lambda}, \beta)$, derived by applying Jensen's inequality [9]:

$$\begin{aligned} \mathcal{L}_q(\mathcal{D}) = \mathbb{E}_q[\ln p(\boldsymbol{\pi} | \alpha)] + \sum_{m=1}^{M} \mathbb{E}_q[\ln p(\theta_m | \boldsymbol{\lambda})] + \sum_{d=1}^{D} \mathbb{E}_q[\ln p(c_d | \boldsymbol{\pi})] \\ + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \mathbb{E}_q[\ln p(w_{d,n} | z_{d,n}, \beta) p(z_{d,n} | \boldsymbol{\theta}, c_d)] - \mathbb{E}_q[\ln q(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z})]. \quad (5) \end{aligned}$$

The optimum is found by setting the partial derivatives with respect to each variational and model parameter to be zero, which corresponds to the variational E-step and M-step, respectively. In the following we separate these equations into two parts and interpret them from the perspective of clustering and projection, respectively.

4.2 Updates for Clustering

As we mentioned in Section 3.2, the specific variables for clustering are document-cluster assignments c_d, cluster centers θ_m, and cluster probabilities $\boldsymbol{\pi}$. It turns out that their corresponding variational parameters are updated as follows:

$$\psi_{d,m} \propto \exp\left\{\sum_{k=1}^{K}\left[\Big(\Psi(\gamma_{m,k}) - \Psi(\sum_{i=1}^{K}\gamma_{m,i})\Big)\sum_{n=1}^{N_d}\phi_{d,n,k}\right] + \Psi(\eta_m) - \Psi(\sum_{i=1}^{M}\eta_i)\right\}, \quad (6)$$

$$\gamma_{m,k} = \sum_{d=1}^{D}\psi_{d,m}\sum_{n=1}^{N_d}\phi_{d,n,k} + \lambda_k, \qquad \eta_m = \sum_{d=1}^{D}\psi_{d,m} + \frac{\alpha}{M}, \quad (7)$$

where $\Psi(\cdot)$ is the digamma function, the first derivative of the log Gamma function. $\psi_{d,m}$ are the *a posteriori* probabilities $p(c_d = m)$ that document d belongs to cluster m, and define a *soft cluster assignment* for each document. $\gamma_{m,k}$ characterize the cluster centers θ_m and are basically the kth coordinate of θ_m on the topic space. Finally η_m control the mixing weights for clusters and define the probability of cluster m. $\phi_{d,n,k}$ are the variational parameters that measure the *a posteriori* probability that word $w_{d,n}$ in document d is sampled from topic k. They are related to projection of words and assumed fixed at the moment.

These equations seem to be complicated and awful, but they turn out to be quite intuitive and just follow the standard clustering procedure. In particular,

- $\psi_{d,m}$ is seen from (6) to be a multiplication of two factors p_1 and p_2, where p_1 includes the γ terms in the exponential and p_2 the η terms. Since η_m controls the probability of cluster m, p_2 acts as a *prior term* for $\psi_{d,m}$; p_1 can be seen as the *likelihood term*, because it explicitly measures the probability of generating $\mathbf{w}_d$ from cluster m by calculating the inner product of projected features and cluster centers. Therefore, (6) directly follows from Bayes' rule, and a normalization term is needed to ensure $\sum_{m=1}^{M}\psi_{d,m} = 1$.
- $\gamma_{m,k}$ is updated by summing over the *prior position* λ_k and the *empirical location*, the weighted sum of projected documents that belong to cluster k.
- Similar to $\gamma_{m,k}$, η_k is empirically updated by summing over the *belongingnesses* of all documents to cluster k. α/M acts as a prior or a *smoothing term*, shared by all the clusters.

Since these parameters are coupled, clustering is done by iteratively updating (6) and (7). Note that the words are incorporated into the clustering process *only* via the projected features $\sum_{n=1}^{N_d}\phi_{d,n,k}$. This means that the clustering is performed not in word space, but in the more informative topic space.

4.3 Updates for Projection

If $\boldsymbol{\psi}, \boldsymbol{\gamma}, \boldsymbol{\eta}$ are fixed, projection parameters $\boldsymbol{\phi}$ and β are updated as:

$$\phi_{d,n,k} \propto \beta_{k,w_{d,n}} \exp\left\{\sum_{m=1}^{M}\psi_{d,m}\Big[\Psi(\gamma_{m,k}) - \Psi(\sum_{i=1}^{K}\gamma_{m,i})\Big]\right\}, \quad (8)$$

$$\beta_{k,j} \propto \sum_{d=1}^{D}\sum_{n=1}^{N_d}\phi_{d,n,k}\delta_j(w_{d,n}), \quad (9)$$

where $\delta_j(w_{d,n}) = 1$ if $w_{d,n}$ takes word index j, and 0 otherwise. Please recall that $\phi_{d,n,k}$ is the *a posteriori* probability that word $w_{d,n}$ is sampled from topic k, and $\beta_{k,j}$ measures the probability of generating word j from topic k. Normalization terms are needed to ensure $\sum_{k=1}^{K} \phi_{d,n,k} = 1$ and $\sum_{j=1}^{V} \beta_{k,j} = 1$, respectively. Update (9) for $\beta_{k,j}$ is quite intuitive, since we just sum up all the documents that word j occurs, weighted by their generating probabilities from topic k. For update of $\phi_{d,n,k}$ in (8), $\beta_{k,w_{d,n}}$ is the probability that topic k generates word $w_{d,n}$ and is thus the *likelihood* term; the rest exponential term defines the *prior*, i.e., the probability that document d selects topic k. This is calculated by taking into account the clustering structure and summing over all cluster centers with corresponding soft weights. Therefore, the projection model is learned via clusters of documents, not simply individual ones. Finally we iterate (8) and (9) until convergence to obtain the optimal projection.

4.4 Discussion

As guaranteed by variational EM algorithm, iteratively performing the given clustering and projection operations will improve the data likelihood monotonically until convergence, where a local maxima is obtained. The convergence is usually very fast, and it would be beneficial to initialize the algorithm using some simple projection models like pLSI.

The remaining parameters α and $\boldsymbol{\lambda}$ control the mixing weights $\boldsymbol{\pi}$ and cluster centers θ_m *a priori*, and they can also be learned by setting their partial derivatives to zero. However, there are no analytical updates for them and we have to use computational methods like Newton-Raphson method as in [2].

The PCP model can also be seen as a Bayesian generalization of the TTMM model [10], where $\boldsymbol{\pi}$ and θ_m are directly optimized using EM. Treating them as variables instead of parameters would bring more flexibility and reduce the impact of overfitting. We summarize the PCP algorithm in the following table:

Table 1. The PCP Algorithm

1. Initialize model parameters $\alpha, \boldsymbol{\lambda}$ and β. Choose $M > 0$ and $K > 0$. Choose initial values for $\phi_{d,n,k}$, $\gamma_{m,k}$ and η_k.
2. **Clustering**: Calculate the projection term $\sum_{n=1}^{N_d} \phi_{d,n,k}$ for each document d and iterate the following steps until convergence:
   (a) Update cluster assignments $\psi_{d,m}$ by (6);
   (b) Update cluster centers $\gamma_{m,k}$ and mixing weights η_k by (7).
3. **Projection**: Calculate the clustering term $\sum_{m=1}^{M} \psi_{d,m} \left[\Psi(\gamma_{m,k}) - \Psi(\sum_{i=1}^{K} \gamma_{m,i})\right]$ for each document d and iterate the following steps until convergence:
   (a) Update word projections $\phi_{d,n,k}$ by (8);
   (b) Update projection matrix β by (9).
4. Update α and $\boldsymbol{\lambda}$ if necessary.
5. Calculate the lower bound (5) and go to Step 2 if not converged.

5 Empirical Study

In this section we illustrate experimental results for the PCP model. In particular we compare it with other models in the following three perspectives:

- **Document Modelling**: How good is the generalization in PCP model?
- **Word Projection**: Is the projection really improved in PCP model?
- **Document Clustering**: Will the clustering be better in PCP model?

We will make comparisons based on two text data sets. The first one is Reuters-21578, and we select all the documents that belong to the five categories *money-fx*, *interest*, *ship*, *acq* and *grain*. After removing stop words, stemming and picking up all the words that occur at least in 5 documents, we finally obtain 3948 documents with 7665 words. The second data set consists of four groups taken from 20Newsgroup, i.e., *autos*, *motorcycles*, *baseball* and *hockey*. Each group has 1000 documents, and after the same preprocessing we get 3888 documents with 8396 words. In the following we use "Reuters" and "Newsgroup" to denote these two data sets, respectively. Before giving the main results, we illustrate one case study for better understanding of the algorithm.

5.1 Case Study

We run the PCP model on the Newsgroup data set, and set topic number $K = 50$ and cluster number $M = 20$. α is set to 1 and $\boldsymbol{\lambda}$ is set with each entry being $1/K$. Other initializations are chosen randomly. The algorithm runs until the improvement on $\mathcal{L}_q(\mathcal{D})$ is less than 0.01% and converges after 10 steps.

Figure 2 illustrates part of the results. In (a) 10 topics are shown with 10 words that have highest assigned probabilities in β. The topics are seen to be very meaningful and each defines one projection for all the words. For instance, topic 5 is about "bike", and 1, 7, 9 are all talking about "car" but with different subtopics: 1 is about general stuffs of car; 7 and 9 are specifying car systems and purchases, respectively. Besides finding topic 6 that covers general terms for "hockey", we even find two topics that specify the hockey teams in US (4) and Canada (8). These topics provide the building blocks for document clustering.

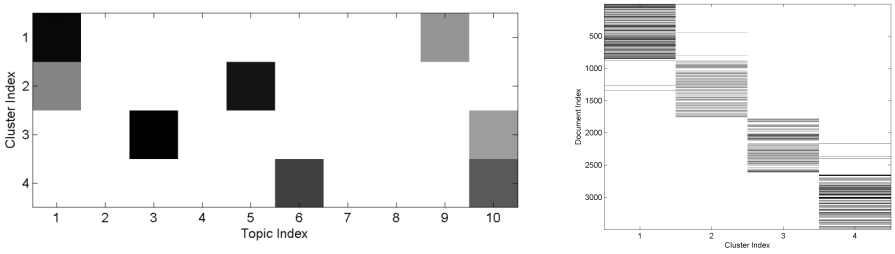
Figure 2(b) gives the 4 cluster centers that have highest probabilities after learning. They define topic mixtures over the whole 50 topics, and for illustration we only show the given 10 topics as in (a). Darker color means higher weight. It is easily seen that they are corresponding to the 4 categories *autos*, *motorcycles*, *baseball* and *hockey*, respectively. If we sort all the documents with their true labels, we obtain the document-cluster assignment matrix as shown in Figure 2(c). Documents that belong to different categories are clearly separated.

5.2 Document Modelling

In this subsection we investigate the generalization of PCP model. We compare PCP with pLSI and LDA on the two data sets, where 90% of the data are used

1	2	3	4	5	6	7	8	9	10
car	ball	game	gm	bike	team	car	pit	car	team
engin	runner	basebal	rochest	clutch	hockei	tire	det	price	year
ford	hit	gant	ahl	back	nhl	brake	bo	dealer	win
problem	base	pitch	st	gear	leagu	drive	tor	year	morri
mustang	write	umpir	john	front	game	radar	chi	model	cub
good	fly	time	adirondack	shift	season	oil	nyi	insur	game
probe	rule	call	baltimor	car	citi	detector	van	articl	write
write	articl	strike	moncton	time	year	system	la	write	jai
ve	left	write	hockei	work	star	engin	stl	cost	won
sound	time	hirschbeck	utica	problem	minnesota	spe	buf	sell	clemen

(a)



(b) (c)

Fig. 2. A case study of PCP model on Newsgroup data. (a) shows 10 topics and 10 associated words for each topic with highest generating probabilities. (b) shows 4 clusters and the topic mixture on the 10 topics. Darker color means higher value. (c) gives the assignments to the 4 clusters for all the documents.

for training and the rest 10% are held out for testing. The comparison metric is *perplexity*, which is conventionally used in language modelling and defined as $\text{Perp}(\mathcal{D}_{\text{test}}) = \exp(-\ln p(\mathcal{D}_{\text{test}})/\sum_d |\mathbf{w}_d|)$, where $|\mathbf{w}_d|$ is the length of document d. A lower perplexity score indicates better generalization performance.

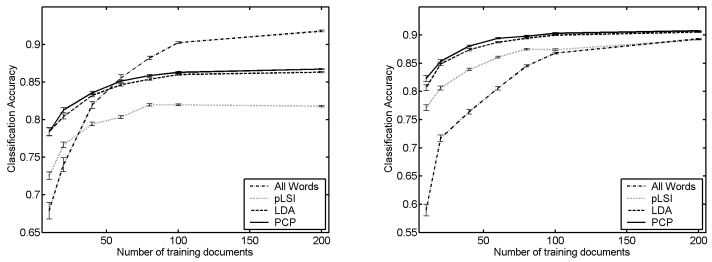
We follow the formula in [2] to calculate perplexity for pLSI. For PCP model, we take the similar approach as in LDA, i.e., we run the variational inference and calculate the lower bound (5) as the likelihood term. M is set to be the number of training documents for initialization. As suggested in [2], a smoothing term for β is used and optimized for LDA and PCP. All the three models are trained until the improvement is less than 0.01%. We compare all three algorithms using different K's, and the results are shown in Table 2. PCP outperforms both pLSI and LDA in all the runs, which indicates that the model fits the data better.

5.3 Word Projection

All the three models pLSI, LDA and PCP can be seen as projection models and learn the mapping β. To compare the quality, we train a support vector machine

Table 2. Perplexity comparison for pLSI, LDA and PCP on Reuters and Newsgroup

	Reuters						Newsgroup					
K	5	10	20	30	40	50	5	10	20	30	40	50
pLSI	1995	1422	1226	1131	1128	1103	2171	2018	1943	1868	1867	1924
LDA	1143	892	678	599	562	533	2083	1933	1782	1674	1550	1513
PCP	**1076**	**882**	**670**	**592**	**555**	**527**	**2039**	**1871**	**1752**	**1643**	**1524**	**1493**



(a) (b)

Fig. 3. Classification results on Reuters (a) and Newsgroup (b).

(SVM) on the low-dimensional representations of these models and measure the classification rate. For pLSI, the projection for document d is calculated as the *a posteriori* probability of latent topics conditioned on d, $p(z|d)$. This can be computed using Bayes' rule as $p(z|d) \propto p(d|z)p(z)$. In LDA it is calculated as the *a posteriori* Dirichlet parameters for d in the variational E-step [2]. In PCP model this is simply the projection term $\sum_{n=1}^{N_d} \phi_{d,n,k}$ which is used in clustering.

We train a 10-topic model on the two data sets and then train a SVM for each category. Note that we are reducing the feature space by 99.8%. In the experiments we gradually improve the number of training data from 10 to 200 (half positive and half negative) and randomize 50 times. The performance averaged over all categories is shown in Figure 3 with mean and standard deviation. It is seen that PCP obtains better results and learns a better word projection.

5.4 Document Clustering

In our last experiment we demonstrate the performance of PCP model on document clustering. For comparison we implement the original version of NMF algorithm [11] which can be shown as a variant of pLSI, and a k-means algorithm that uses the learned features by LDA. For NMF we tune its parameter to get best performance. The k-means and PCP algorithms are run with the true cluster number, and we tune the dimensionality K to get best performance.

The experiments are run on both two data sets. The true cluster number is 5 for Reuters and 4 for Newsgroup. For comparison we use the normalized mutual information [13], which is just the mutual information divided by the maximal entropy of the two cluster sets. The results are given in Table 3, and it

Table 3. Comparison of clustering using different methods

	NMF	LDA+k-means	PCP
Reuters	0.246	0.331	**0.418**
Newsgroup	0.522	0.504	**0.622**

can be seen that PCP performs the best on both data sets. This means iterating clustering and projection can obtain better clustering structure for documents.

6 Conclusions

This paper proposes a probabilistic clustering-projection model for discrete co-occurrence data, which unifies clustering and projection in one probabilistic model. Iteratively updating the two operations turns out to be the variational inference and learning under Bayesian treatments. Experiments on two text data sets show promising performance for the proposed model.

References

1. C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
2. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
3. W. Buntine and S. Perttu. Is multinomial PCA multi-faceted clustering or dimensionality reduction? In *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics*, pages 300–307, 2003.
4. S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
5. I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *SIGKDD*, pages 269–274, 2001.
6. C. Ding, X. He, H. Zha, and H. D. Simon. Adaptive dimension reduction for clustering high dimensional data. In *ICDM*, pages 147–154, 2002.
7. T. Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual ACM SIGIR Conference*, pages 50–57, Berkeley, California, August 1999.
8. T. Hofmann and J. Puzicha. Statistical models for co-occurrence data. Technical Report AIM-1625, 1998.
9. M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.
10. M. Keller and S. Bengio. Theme Topic Mixture Model: A Graphical Model for Document Representation. January 2004.
11. D. D. Lee and H. S. Seung. Learning the parts of objects with nonnegative matrix factorization. *Nature*, 401:788–791, Oct. 1999.
12. T. Li, S. Ma, and M. Ogihara. Document clustering via adaptive subspace iteration. In *Proceedings of SIGIR*, 2004.
13. W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of SIGIR*, pages 267–273, 2003.