Decomposing Background Topics from Keywords by Principal Component Pursuit

Kerui Min[1] Zhengdong Zhang[2] John Wright[3] Yi Ma[3]
[1]School of Computer Science, Fudan University, Shanghai, China
[2]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[3]Microsoft Research Asia, Beijing, China
[1]krmin@fudan.edu.cn, [2]zhangzd07@mails.tsinghua.edu.cn, [3]{jowrig,mayi}@microsoft.com

ABSTRACT

Low-dimensional topic models have been proven very useful for modeling a large corpus of documents that share a relatively small number of topics. Dimensionality reduction tools such as Principal Component Analysis or Latent Semantic Indexing (LSI) have been widely adopted for document modeling, analysis, and retrieval. In this paper, we contend that a more pertinent model for a document corpus as the combination of an (approximately) low-dimensional topic model for the corpus and a sparse model for the keywords of individual documents. For such a joint topic-document model, LSI or PCA is no longer appropriate to analyze the corpus data. We hence introduce a powerful new tool called Principal Component Pursuit that can effectively decompose the low-dimensional and the sparse components of such corpus data. We give empirical results on data synthesized with a Latent Dirichlet Allocation (LDA) mode to validate the new model. We then show that for real document data analysis, the new tool significantly reduces the perplexity and improves retrieval performance compared to classical baselines.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Theory, Verification, Experimentation

Keywords

Principal Component Pursuit, Sparse Keywords, Latent Semantic Indexing, Latent Dirichlet Allocation, Perplexity

This work is partially supported by the grants NSF CCF 0964215 and ONR N000140910230

CIKM'10, October 26–29, 2010, Toronto, Ontario, Canada.

1. INTRODUCTION

The ability to effectively model and analyze relationships among a large corpus of documents has become increasingly important for information retrieval, and for indexing, ranking, and searching documents on the web [7, 15]. In the past few years, a large number of models and computational tools have been introduced in the information retrieval literature, and many have seen great success and led to more powerful indexing and search methods.

Nevertheless, the rapidly increasing volume and diversity of the web documents continue to demand ever more effective and scalable tools for extracting useful information from massive collections of text data. The size of the document corpus to be analyzed is now routinely in the range of millions or even billions. When dealing with such massive high-dimensional data, the *curse of dimension* becomes a reality: the complexity of many models and algorithms increases exponentially in terms of the dimension, or their performance decreases sharply as the scale goes up. As pointed out in [3], to alleviate such a curse, modern statistical and computational methods have started to leverage on the fact that natural high-dimensional data typically have low intrinsic dimensionality.

Such low-dimensional structures have long been observed and harnessed in the analysis of document corpora. Although the number of documents in a corpus can be large, their topics can be very limited and highly correlated. Thus, if we try to summarize the semantics of the corpus at the topic level, the corpus data could have much lower dimension (or degree of freedom) than the ambient dimension (the number of documents). Since such a low-dimensional model is important for summarizing the semantics of the corpus, earlier work such as the *Latent Semantic Indexing* (LSI) [8] has proposed to use *Singular Value Decomposition* (or *Principal Component Analysis*) to fit a low-dimensional subspace to the corpus data matrix. This simple subspace model and solution by PCA have had tremendous influence on document analysis in the past two decades.

Many recent more powerful (and hence more complicated) models such as *Probabilistic Latent Semantic Indexing* (pLSI) [10] and *Latent Dirichlet Allocation* (LDA) [2] essentially follow the same rationale that document corpora at the topic level have few intrinsic degrees of freedom. Although pLSI and LDA are based on more precise statistical formulations, in practice, data generated according to such models clearly exhibit an approximate low dimensional subspace structure, as shown in Figure 3(a). Mathematically, the corpus data matrix D can be viewed as low-rank matrix L_0 perturbed

CHRYSLER SETS STOCK **SPLIT**, HIGHER **DIVIDEND**

Chrysler Corp said its board declared a three-for-two stock **split** in the form of a 50 pct stock **dividend** and raised the quarterly **dividend** by seven pct.

The company said the **dividend** was raised to 37.5 cts a share from 35 cts on a pre-split basis, equal to a 25 **ct dividend** on a **post-split** basis.

Chrysler said the stock **dividend** is **payable** April 13 to holders of record March 23 while the cash **dividend** is **payable** April 15 to holders of record March 23. It said cash will be paid in lieu of fractional shares.

With the **split**, **Chrysler** said 13.2 mln shares remain to be purchased in its stock repurchase program that began in late 1984. That program now has a target of 56.3 mln shares with the latest stock **split**.

Chrysler said in a statement the actions "reflect not only our outstanding performance over the past few years but also our **optimism** about the company's future."

U.S. COMMERCE'S **ORTNER** SAYS **YEN UNDERVALUED**

Commerce Dept. undersecretary of economic affairs Robert **Ortner** said that he believed the dollar at current levels was fairly priced against most European currencies.

In a wide ranging address sponsored by the Export-Import Bank, **Ortner**, the bank's senior economist also said he believed that the **yen** was **undervalued** and could go up by 10 or 15 pct.

"I do not regard the dollar as **undervalued** at this point against the **yen**," he said.

On the other hand, **Ortner** said that he **thought** that "the **yen** is still a **little** bit **undervalued**," and "could go up another 10 or 15 pct."

In addition, **Ortner**, who said he was speaking personally, said he **thought** that the dollar against most European currencies was "fairly priced."

Ortner said his analysis of the various exchange rate values was based on such economic particulars as wage rate differentiations.

Ortner said there had been **little impact** on U.S. trade deficit by the decline of the dollar because at the time of the Plaza Accord, the dollar was **extremely** overvalued and that the first 15 pct decline had **little impact**.

He said there were indications now that the trade deficit was beginning to level off.

Turning to Brazil and Mexico, **Ortner** made it clear that it would be almost impossible for those countries to earn enough foreign exchange to pay the service on their debts. He said the best way to deal with this was to use the policies outlined in Treasury Secretary James Baker's debt initiative.

Figure 1: Highlighted keywords in two sample texts from the Reuters dataset. The bold words are keywords contained in the sparse term S, while black words are either not in the vocabulary or are explained by the low-rank model L. Here, the corpus was decomposed via PCP with $\lambda = 5/\sqrt{3,000}$.

by small noise:

$$D = L_0 + Z_0. \tag{1}$$

To some extent, pLSI and LDA model use generative statistical models to justify the validity of PCA for document data analysis: if Z_0 is i.i.d. Gaussian, PCA in fact gives the optimal estimate of the low-dimensional subspace (or principal components) L_0.

While most of the above topic models are trying to capture the intrinsic common relationships among all the documents, none of them is designed to model the statistics of individual documents. Although a particular document may belong to a common topic (say on the topic of "finance"), it covers a story different from all the other financial documents in the same corpus. Any statistical deviation of this document from the common topic model allows us to distinguish it from the rest of the corpus. For instance, certain words may show up much more frequently in the document and stand out from the background topic model. One may view such words as "keywords." Figure 1 shows one example of some such keywords detected by our method in a financial document. Obviously, discriminative statistical features like the keywords would be tremendously informative for us to identify any particular document inside the entire corpus, hence allow us to better index, search and rank documents [18].

In this paper, we propose to model the statistics of both the corpus topic and individual documents with a combination of a topic model such as LDA and a sparse model for the keywords that occur with unusually high frequency in each document. We refer to such a model as a *joint topic-document model*. Similar models have previously been contemplated and explored for document analysis [5]. However, the solution given in [5] suffers from several problems. First, [5] directly models the keywords and topic words via a latent random variable and estimates the parameters from the corpus. The definition of keywords, however, is more a subjective rather than objective concept. What percentage of words should be treated as the keywords of a document, 10%, 5%, or 1%? The answer varies from application to application. Our model leaves it as a free parameter to tune, embracing a broader range of applications. Moreover, although the inference procedure (a modified Gibbs Sampling method) used by [5] worked well in practice, there is no guarantee on the number of iterations in general.

As we will soon see, mathematically, we essentially assume that the corpus data can be written as the sum of a low-rank matrix and a sparse matrix

$$D = L_0 + S_0 \tag{2}$$

with the low-rank component L_0 corresponding to the background topic and the sparse one S_0 to the keywords (here, we assume D, L_0 and S_0 to be unnormalized term frequency matrixes). There is a major difference between this model and the LSI model discussed earlier: there Z_0 (given in (1)) introduces an entry-wise small perturbation to L_0 while here the entries of S_0 are sparse but in theory are allowed to have arbitrarily large magnitude. It is well known that decomposing a matrix into its low-rank and sparse components is an intractable (NP-hard [4]) problem in general. So such a model, though conceptually well-motivated, would be useless for real document analysis if we could not effectively learn the low-dimensional background topic and the sparse keywords from large corpora. Fortunately, recent breakthroughs in high-dimensional convex optimization indicate that the above decomposition can be exactly and efficiently computed under surprisingly broad conditions by solving a certain convex program, called *Principal Component Pursuit* (PCP) [3].

In this paper, we validate the joint topic-document model with encouraging empirical results on synthetic and real data. Our experiments demonstrate the effectiveness of PCP in recovering the low-dimensional topic model and the sparse keywords. As we will see, this new tool yields much better results in terms of both subspace distance and perplexity in identifying the latent low-dimensional topic model, and these translate into better performance in identifying keywords as well as on a model retrieval task.

We note that the goal of this paper is *not* to reach at a full-fledged document indexing and retrieval method that has been optimized to work better than existing methods on diverse corpora. Instead, our goal is to validate a simple, new model for document analysis and to introduce a remarkable new tool that can learn such new model rather effectively. Although we have only verified the model on a corpus of moderate size and compared with classical baseline methods, the consistently positive test results and wide improvement margins indicate that this new model indeed has great potential to enhance future document analysis, indexing, and retrieval.

2. JOINT TOPIC-DOCUMENT MODEL

In this section, we introduce the proposed joint topic-document model and new computational tool for analyzing corpus data that obey this model. As the new model is a generalization of conventional topic models, especially LSI and LDA, for completeness, we first give a brief overview of main assumptions of these methods. The overview will also discuss some limitations of these assumptions and justify the need for a better model and analytical tool.

2.1 Low-dimensional Background Topics

It is common in document analysis to assume that documents are generated from a relatively small number of topics [1, 2, 8, 10]. We consider documents containing words drawn from a vocabulary $\mathcal{V}$ of size m. Each topic corresponds to a discrete probability distribution over $\mathcal{V}$. To fix some notation, suppose there are r topics, and that $\tau_1 \dots \tau_r$ denote these r distributions: each $\tau_i \in \mathbb{R}^m$ is a nonnegative vector whose entries sum to one. For ease of notation, we introduce a matrix

$$T \;=\; [\tau_1 \mid \cdots \mid \tau_r] \;\in\; \mathbb{R}^{m\times r}. \tag{3}$$

We will let n denote the number of documents in the corpus. For each $j = 1 \dots n$, the j-th document is generated as follows: one first chooses a set of r weights, $w^{(j)} = (w_1^{(j)} \dots w_r^{(j)}) \in \mathbb{R}^r$, which are nonnegative and sum to one. One then forms the mixture distribution

$$p^{(j)} \;=\; \sum_i w_i^{(j)} \tau_i \;=\; T w^{(j)}.$$

Clearly, $p^{(j)}$ is also a probability distribution over $\mathcal{V}$. The words of the j-th document are considered as an iid sample of length N_j from $p^{(j)}$: $s^{(j)} = (s_1^{(j)} \dots s_{N_j}^{(j)}) \sim_{iid} p^{(j)}$. The final observation is the unnormalized term frequency vector: $d^{(j)} \in \mathbb{R}^m$ with

$$d^{(j)}(i) \;=\; \#\{k \mid s_k^{(j)} = i\}$$

. In particular, if the length of the document N_j is large, then the law of large numbers suggests that with proper normalization, the empirical probability distribution

$$d^{(j)}/N_j \;\approx\; p^{(j)}.$$

It is convenient to further consider the matrix forms

$$\begin{aligned} W &= \left[w^{(1)} \mid \cdots \mid w^{(n)} \right] \;\in\; \mathbb{R}^{r\times n}, \\ P &= \left[p^{(1)} \mid \cdots \mid p^{(n)} \right] \;\in\; \mathbb{R}^{m\times n}, \\ D &= \left[d^{(1)} \mid \cdots \mid d^{(n)} \right] \;\in\; \mathbb{R}^{m\times n}. \end{aligned}$$

Here $D \in \mathbb{R}^{m\times n}$ is our observation (again, note that T, W and P are all probability or weight matrices, but D is unnormalized term frequency matrix "sampled" from P). In terms of these quantities, notice that

$$P = TW, \quad \text{and} \quad \operatorname{rank}(P) = r.$$

Since column-wise normalization does not affect the rank of D, D is approximately the same rank as P (i.e., r) except for the small noise term and lack of sufficient sampling. This phenomenon has been verified empirically in real text corpora. For example, for a document set consisting of $n = 10,000$ articles with a vocabulary $\mathcal{V}$ of size $m = 20,000$, between 50 and 300 dimensions are generally sufficient [6, 11]. This has motivated the use of low-rank approximation in information retrieval, under the name *Latent Semantic Indexing* [8]. LSI uses the singular value decomposition to form an optimal rank-r approximation

$$\hat{L}_{LSI} \;=\; \sum_{i=1}^{r} u_i \sigma_i v_i^T, \tag{4}$$

where $\{u_i\}$, $\{v_i\}$ are the first r singular vectors of D and $\{\sigma_i\}$ the corresponding singular values. Interestingly, in some cases this low-rank approximation indeed improves recall [8], a phenomenon that can be explained under certain strong generative models for text [16].

Although the above low-dimensional model seems naive, it captures the essence of other more sophisticated topic models such as the popular LDA model. Although not explicitly analyzed in [2], one can show that if each document to be generated by LDA model is sufficiently long, the mixture weights of topics of T can be computed accurately. Therefore, the corpus generated by LDA model satisfies the low-dimension model. In summary, LDA places probabilistic priors on the number of words per document and the mixing weights, which makes it more flexible in modeling real

corpus data. Hence, in subsequent simulations and experiments in Section 3, we will use the LDA to generate and model our data. We also verified empirically the property of low-dimensionality of LDA here.

Note that in the following discussion, unless explicitly stated, D, L, S and all the related matrix symbols all stands for document-term matrixes whose entries indicate the number of occurrence of each term.

2.2 Sparse Model for Keywords

As argued above, a low-dimensional topic model (and its low-rank approximation) provides a good representation of the *commonalities* of a set of documents. However, for indexing and retrieval tasks, the deviation of each document from the common model may actually be much more informative. Normally each document will have a few terms that are used with much higher frequency than one might expect from the overall statistics of the corpus. We call these the "keywords" of the document, and model their effect on the observation D through an additive term $S_0 \in \mathbb{R}^{m\times n}$:

$$D \;\approx\; L_0 + S_0. \tag{5}$$

Since there are relatively few keywords, S_0 is a *sparse* matrix.

In the context of document analysis, the above model is rather natural to interpret: while the low-rank component L_0 can be viewed as the common "background" topics of all the documents in the corpus; the sparse component S_0 capture the distinctive "keywords" or key phrases that best represent the unique content of each document, see for example the words "Chrysler" and "Ortner" in Figure 1. The support of S_0 indicates which terms are the keywords for each document and the magnitude of its entries suggests how much each keyword stand out from the common topic model. As this model capture both the corpus topic and the content of individual documents, we call it a "joint topic-document" model.

This model has several implications. First, as discussed above

> An approximation to the low-rank component L_0 may not contain the most relevant information for indexing.

Moreover,

> Under this model, SVD-based methods may not even compute a good approximation L_0.

The reason for this is simple: the rank-r approximation (4) is optimal when the matrix D is perturbed by small dense noise, say $D = L_0 + Z_0$ where Z_0 is i.i.d. Gaussian. However, the sparse perturbation S_0 due to the presence of keywords has a very different nature than Gaussian noise: a few of the entries are quite large, but the rest are almost zero. In this situation of large but sparse corruption, the approximation error $\|\hat{L}_{LSI} - L_0\|$ can become extremely large.

2.3 Decomposing Background Topics and Keywords

It seems then, that a more pertinent solution for the joint topic-document model would be to attempt to directly compose the data matrix D in (5) into a low-rank component L_0, corresponding to the topic model, and a sparse component S_0, corresponding to the keywords. Clearly, we can always get a low-rank L_0 by setting $S_0 = D$, which is obviously meaningless. Therefore, we want to minimize the number of non-zero entries in S_0 so long as L_0 is a low-rank matrix. Specifically if we know from prior knowledge that the estimated number of topics of the given corpus is at most r, the above idea can be translated into the following optimization problem:

$$\min \|S\|_0 \quad \text{subject to} \quad L + S = D, \text{rank}(L) \le r \tag{6}$$

The Lagrange form of (6) is:

$$\min \text{rank}(L) + \lambda\|S\|_0 \quad \text{subject to} \quad L + S = D \tag{7}$$

It is known, however, for worst case inputs, this optimization problem is intractable (NP-hard [4]). Nevertheless, as we will see, recent advances in the study of low-rank matrix recovery provide a computationally feasible solution that guarantees to solve a surprisingly broad class of instances. In fact, these advances are largely motivated by the interest in relevance analysis of web data or user taste anticipation. In particular, as long as the rank of the matrix L_0 is sufficiently low and the matrix S_0 is sparse, one can effectively and efficiently get *exactly the same* solution as (7) by solving the following relaxed convex program:

$$\min \|L\|_* + \lambda\|S\|_1 \quad \text{subject to} \quad L + S = D. \tag{8}$$

Here $\|\cdot\|_*$ is the nuclear norm of a matrix (i.e. the sum of its singular values) and $\|\cdot\|_1$ is the ℓ_1 norm of a matrix (i.e. the sum of absolute values of entries). The parameter $\lambda > 0$ balances the tradeoff between sparsity of the keywords matrix S and rank of the background topic matrix L. According to the theoretical result established in [3], λ should be of the order of $\Theta\left(\max\{m,n\}^{-1/2}\right)$. The above convex programming problem is dubbed *Principal Component Pursuit (PCP)* in [3]. Below, we will let

$$\hat{L}_{PCP},\ \hat{S}_{PCP} \tag{9}$$

denote the solution to (8). The recent development in convex optimization has produced algorithms that can solve this relaxed convex program with a computational cost not so much higher than that of the classical PCA. For completeness we review one of the fastest algorithm via *Alternating Directions* [19, 20] method here to demonstrate the tractability of (8). The equation (8) can be rewritten as the Augmented Lagrange Multiplier form:

$$\begin{aligned}\min \ell(L,S,Y) = \|L\|_* + \lambda\|S\|_1 + \langle Y, D - L - S\rangle \\ + \frac{\mu}{2}\|D - L - S\|_F^2\end{aligned} \tag{10}$$

where the Euclidean inner product between two matrices X and Y is defined as $\langle X, Y\rangle = \text{trace}(X^*Y)$.

There are two important observations that makes minimizing (10) quite easy and elegant. Before presenting that, let us define two operators:

$$\mathcal{S}_\tau(x) = \text{sgn}(x)\max(|x| - \tau, 0)$$

and

$$\mathcal{D}_\tau(X) = U\mathcal{S}_\tau(\Sigma)V^*, X = U\Sigma V^*$$

Here $\mathcal{S}_\tau(\cdot)$ is initially defined on single number and then we extend it to matrix by applying it to every entry in the

The recent Netflix challenge of completing movie rankings from incomplete user survey is one such example.

matrix. For any matrix X, $X = U\Sigma V^*$ gives the singular value decomposition of it. The following two observations are particularly important for the minimization problem:

$$\arg\min_S \ell(L, S, Y) = \mathcal{S}_{\lambda\mu^{-1}}(D - L + \mu^{-1}Y) \quad (11)$$

$$\arg\min_L \ell(L, S, Y) = \mathcal{D}_{\mu^{-1}}(D - S_k - \mu^{-1}Y_k) \quad (12)$$

With these two observations, in each iteration we fix S and minimize ℓ with respect to L, do the contrary(fix L and minimize with respect to S) and then update the Lagrange multiplier matrix Y based on the residual $D - L - S$. This procedure has been proved to converge to the global optimal point of (10) under quite board conditions [19, 20]. The above idea is summarized as Algorithm 1.

Algorithm 1 Decomposing D into Low-rank and Sparse Parts by Alternating Directions [19, 20]

Initialize: $S_0 = Y_0 = 0$, $\mu > 0$
while not converged **do**
  Compute $L_{k+1} = \mathcal{D}_{\mu^{-1}}(D - S_k - \mu^{-1}Y_k)$
  Compute $S_{k+1} = \mathcal{S}_{\lambda\mu^{-1}}(D - L_{k+1} + \mu^{-1}Y_k)$
  Compute $Y_{k+1} = Y_k + \mu(D - L_{k+1} - S_{k+1})$
end while
return L, S

2.4 Computational Cost and Scalability of Decomposition

As we have discussed, realistic web corpora may contain millions or even billions of observations. This scale is often beyond the capability of traditional methods, due to the *Curse of Dimensionality*. On the other hand, theoretical results in [3] demonstrate that as the data dimensionality increases, the decomposing ability of (8) becomes identical to (7) and perhaps more surprisingly, the convergence of Algorithm 1 needs even fewer iterations, arguably a *Blessing of Dimensionality!*

We will talk about the computational cost of Algorithm 1 briefly. Note that the Singular Value Decomposition is required in each iteration. Recent research shows that solving (8) with this algorithm just takes approximately 10 Singular Value Decomposition on D [20], meaning that the time complexity of solving (8) is almost the same as traditional Singular Value Decomposition, which can be efficiently computed in parallel. Our distributed implementation of Algorithm 1 performs efficiently even for large scale problems, *e.g.*, decomposing matrices as large as $10,000 \times 50,000$.

As we have seen, PCP offers a way to make the conventional PCA robust to gross corruptions with small additional computational cost. In the next section, we will see that this approach performs well in decomposing corpera D that are generated according to the above joint topic-document model.

3. SIMULATIONS AND EXPERIMENTS

In this section, we evaluate the proposed model and algorithm with both synthetic and real data experiments. We first show that if the corpus is generated from a topic model plus sparse document-specific keywords, Principal Component Pursuit gives a much more accurate estimate of L_0 (and hence of S_0) than classical PCA/LSI. We observe that real data such as the Reuters dataset indeed exhibit a distribution that better resembles the joint topic-document model than the topic model alone. Motivated by this observation, we then compare topic models fit to the original data matrix to topic models fit to the matrix after PCP has removed outlying words. We will see that such preprocessing significantly reduces the perplexity of the topic model fit by LDA. Finally, we show how the low-rank and sparse components learned by the PCP algorithm can enhance retrieval performance.

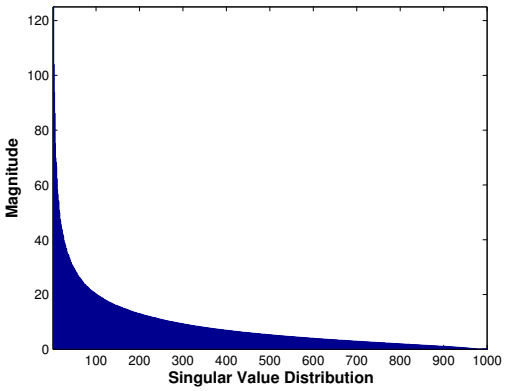


Figure 2: Singular values of the 1,000 documents from the Reuters dataset. The singular values decay smoothly, similarly to the idealized joint topic-document model in Figure 3(b).

3.1 Validation with Synthetic Data

We first perform a basic consistency check, to see that when the corpus is indeed generated according to a topic model plus some sparse keywords, Principal Component Pursuit effectively estimates the two components.

We adopt the popular Latent Dirichlet Allocation model to generate the corpus. This model places probabilistic priors on the number of words N_j per document and the mixing weights $w^{(j)}$. In particular, the N_j are assumed to be i.i.d. samples from a Poisson(ξ) distribution, while the $w^{(j)}$ are assumed to be i.i.d. samples from a Dirichlet distribution with parameter vector $\alpha \in \mathbb{R}^r$. For more details on this probabilistic formulation, we invite the reader to consult [2, 17].

In our simulation, we generate data according to the LDA model with expected document length $\xi = 500$ and Dirichlet parameter $\alpha = (0.1, \ldots, 0.1)$. The corpus size is $n = 1,000$, while the vocabulary is assumed to have size $m = 2,000$. We generate $r = 100$ topics, each distributed in accordance with *Zipf's Law*, which was observed on almost any corpus that is large enough [9]. For each τ_i, we generate an independent random permutation of π_i of $\{1 \ldots m\}$, and set $\tau_i(j) \propto 1/\pi_i(j)$. This is reasonable, since only a small portion of the vocabulary set is highly correlated with a given topic.

We first generate a corpus D_{LDA} according to the LDA model, with no additional keywords. Figure 3(a) plots the singular values of a document matrix generated according to this model. We observe a clear sudden drop of magnitude of the r-th and $(r+1)$-th singular value, where r corresponds to the number of hidden topics. Such a distribution clearly

supports the use of subspace methods such as LSI for processing the data to help identifying the topics.

However, real text datasets rarely exhibit such a clear subspace structure. Figure 2 shows the singular value distribution of a 1,000-article real dataset from the Reuters corpus. As we see, the change of singular values is more gradual and smooth, and one does not see any clear break point.

We next consider what happens if the corpus contains additional keywords, that occur more frequently than predicted by the LDA model. From each article in the synthetically generated corpus, we randomly select 10% of the words of each article to be keywords. We add a frequency of 10 to each (recall that since $\xi = 500$, this is 1/50 of the expected document length), forming a new data matrix

$$D_{JTD} = D_{LDA} + S_0 \approx L_0 + S_0.$$

Figure 3(b) shows the distribution of singular values of the corrupted LDA data. Notice that the clear break point has disappeared and the distribution has become just as smooth as the real data. This shows that the joint topic-document model is more suitable than the LDA model alone, which does not take the keywords of individual documents into consideration.

We next apply principal component pursuit to the perturbed corpus, with weight factor $\lambda = 0.1$. This yields a low-rank and sparse pair $(\hat{L}, \hat{S})$, for which $\hat{S}$ has 54,704 non-zero entries. Figure 3(c) plots the singular values of the recovered matrix $\hat{L}$. Notice that the distribution of singular values becomes much closer to the original distribution in Figure 3(a). The sharp drop at $r = 100$ which vanished due to the presence of keywords is now clearly visible again.

We evaluate the recovery quantitatively by examining the distance between the range of the recovered low-rank subspace and the range of the matrix T of topic distributions defined in (3). Notice that as the number of words per document becomes large, the matrix D_{LDA} approaches $\text{range}(T) = \text{range}(L_0)$. If our goal is to infer the underlying topic mixtures $L_0 = TW$ that generated the data, then the distance to $\text{range}(L_0)$ is a good measure of correctness.

We measure distance using the *subspace angles* [14]. Specifically, let S_1 and S_2 be two subspaces in $\mathbb{R}^n$ with

$$\max\{\dim S_1, \dim S_2\} \leq l.$$

The distance between S_1 and S_2 is defined as follows

$$d(S_1, S_2) = \left(\sum_{i=1}^{l} \theta_i \right)^{1/2}, \tag{13}$$

where $\cos\theta_i = \sigma_i$ and σ_i is the ith singular value of $S_1^T S_2$.

We let $\hat{L}_{LDA,LSI}$ denote the rank r approximation to the LDA corpus D_{LDA} obtained by singular value decomposition. We let $\hat{L}_{JTD,LSI}$ denote the rank r approximation to D_{JTD} obtained by singular value decomposition. Finally, $\hat{L}_{JTD,PCP}$ denotes the rank-r approximation given by applying Principal Component Pursuit to the corpus D_{JTD}. Table 1 plots the distance between the range of each of these recovered low-rank components and the range of L_0. Notice that in the absence of additional sparse keywords, the SVD (or LSI) gives a fairly good approximation to L_0:

By abuse of notion, we here use the same notation S_1, S_2 to represent any orthonormal bases for the two subspaces, respectively.

Subspace	Distance
$d(\text{range}(\hat{L}_{LDA,LSI}), \text{range}(L_0))$	1.86
$d(\text{range}(\hat{L}_{JTD,LSI}), \text{range}(L_0))$	11.16
$d(\text{range}(\hat{L}_{JTD,PCP}), \text{range}(L_0))$	**3.63**

Table 1: The distance between different subspaces.

$d(\text{range}(\hat{L}_{LDA,LSI}), \text{range}(L_0)) \approx 1.86$. However, when additional sparse keywords are introduced this estimate breaks down: $d(\text{range}(\hat{L}_{JTD,LSI}), \text{range}(L_0)) \approx 11.1$. If we instead decompose the data using principal component pursuit, the error drops significantly: $d(\text{range}(\hat{L}_{JTD,PCP}), \text{range}(L_0)) \approx 3.63$.

This drastic reduction clearly suggests that PCP could effectively isolate sparse keywords that cause deviations from the low-rank topic model. Conversely, the large error in estimating L_0 using LSI/SVD suggests that these tools may be less appropriate for such perturbed corpora. We have repeated this experiment with varying numbers of topics, documents and perturbation percentages. The reduction of the subspace angle is consistent and significant.

3.2 Reducing Model Perplexity of Real Data

In this section, we further validate the proposed model with experiments on real data. Because real data lack ground truth L_0, [2] proposed to measure the quality of a learned topic model through the *perplexity*

$$\text{perplexity}(D \mid \text{model}) = \exp\left\{ -\frac{\sum_{j=1}^{n} \log \mathbb{P}(d^{(j)} \mid \text{model})}{\sum_{d=1}^{M} N_d} \right\}.$$

Since this is a monotonic function of the likelihood of D, lower perplexity implies higher likelihood, suggesting a better fit.

3.2.1 Experiment Setup

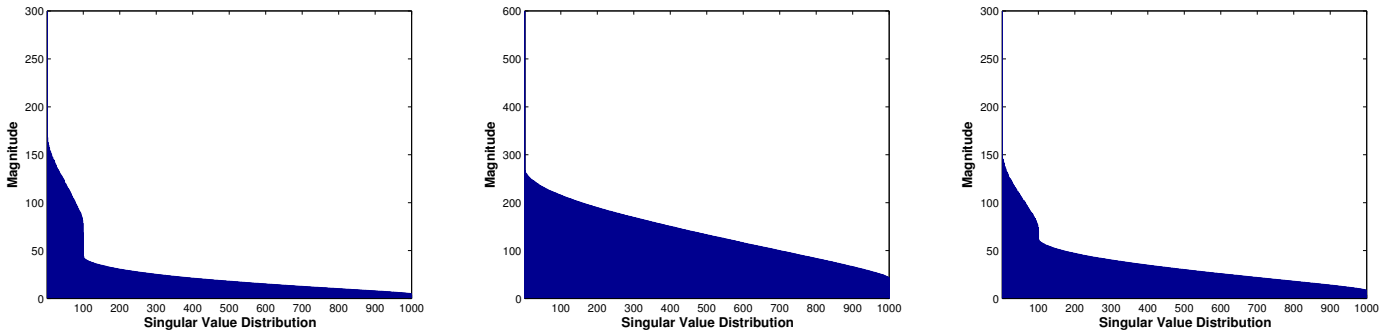
In this experiment, we use a corpus consisting of 1,000 documents in the Reuters-21578 dataset. The vocabulary $\mathcal{V}$ consists of the 3,000 most frequent words, excluding common words in a "stop words" list. This gives us a $3,000 \times 1,000$ data matrix D whose i, j entry is the frequency of occurrence of word i in document j. As above, we decompose D into low-rank topics L and sparse keywords S by solving the Principal Component Pursuit problem (8).

The free parameter λ in (8) strikes a balance between extracting more keywords and using a higher-rank topic model. In our experiment we try 20 different values of λ,

$$\lambda \in \left\{ \frac{1}{\sqrt{3,000}}, \ldots, \frac{20}{\sqrt{3,000}} \right\}.$$

This gives 20 solutions $(L^{(i)}, S^{(i)}), 1 \leq i \leq 20$. For each $L^{(i)}$, we train an LDA model using Gibbs sampling. We let number of topics $r = 50$, and use the default hyperparameters $\alpha = \frac{50}{r} = 1, \beta = 0.1$ and number of iterations, $2,000$. This yields 20 different learned LDA models. For comparison, we also generate an LDA model from the data itself with no PCP preprocessing. We compare the quality of these learned models by evaluating the likelihood of D being generated by the learned model, or, equivalently, the perplexity.

Using the package GibbsLDA++, available at `http://gibbslda.sourceforge.net/`



(a) Synthetic Topic Data (LDA model) (b) Topic Data with Sparse Keywords (c) Low rank component $\hat{L}$ by PCP

Figure 3: Singular values of synthetic topic data. Left: singular values of synthetic topic data sampled according the LDA model with $r = 100$ topics. Notice the sharp drop at 100. Middle: the singular value distribution of the LDA data with additional sparse keywords. The singular value distribution now decays smoothly, with no sharp drop. Right: The singular value distribution of the matrix $\hat{L}$ recovered by applying PCP to the corrupted LDA matrix. Notice that the low-rank structure is recovered: the sharp drop at $r = 100$ is again clearly visible.

We calculate the perplexity of this model and compare those of the models learned from above. Here we ignore the result from $L^{(1)}$ to $L^{(4)}$ and $L^{(19)}$ to $L^{(20)}$. The reason is when λ is too small ($i < 5$), S is too dense containing too many words; and when λ is large ($i > 18$), S becomes too sparse (less than 1,000 entries, which means each document has less than 1 keyword on average). The perplexity as a function of λ is plotted in Figure 4(a), and compared to the baseline perplexity (red line) without PCP.

From the result we see that in the selected range of λ, the perplexity of the model after processed by PCP is significantly lower than that of the unprocessed data. Note that the perplexity for $L^{(5)}$ and $L^{(6)}$ is much lower than that for the original data D, with a reduction of nearly $20 - 30\%$ in the perplexity value. The matrix $S^{(6)}$ is already rather sparse: it contains no more than 150 words compared with the $3,000$ in total. Figure 1 shows some of the keywords (in bold) detected by S in two typical sample texts from the corpus.

From this experiment, we may conclude that after applying PCP to the corpus data and removing a small number of outlying words from each documents, the processed corpus fit the LDA model much better.

3.2.2 *Generalizability Test*

Next, we would like to test the generalizability of the learned model for new (but similar) test data. Again, we use perplexity to measure the goodness of each model learned in explaining the new data. Since the learned LDA model is only supposed to predict the low-dimensional part of the test data, we will first run PCP on the test to remove outliers.

We use the same data set, Reuters-21578 with 2,000 documents, obtaining a 3,000-by-2000 matrix D. But this time we take the odd columns and form a 3,000-by-1,000 submatrix as the training data D_{train} and use the submatrix of the even columns as the testing data D_{test}. For both D_{train} and D_{test}, we use PCP to decompose them to their low rank plus sparse components:

$$\begin{aligned} D_{train} &= L_{train} + S_{train}, \\ D_{test} &= L_{test} + S_{test}. \end{aligned}$$

Same as the previous experiment, we choose 20 different λ from $\frac{1}{\sqrt{3,000}}$ to $\frac{20}{\sqrt{3,000}}$ with step $\frac{1}{\sqrt{3,000}}$. Then we get $L^{(i)}_{train}, L^{(i)}_{test}, 1 \le i \le 20$.

Then we learn an LDA model for each $L^{(i)}_{train}$ with exactly the same setting as before. Also we use the original unprocessed corpus D_{train} to learn an LDA model and compare its perplexity with those learned from $L^{(i)}_{train}$.

Then we would like to test the models on the test matrix M_{test} and $L^{(i)}_{test}, 1 \le i \le 20$. In this inference step we set $iter = 20$ by default. We then calculate the perplexity in the same manner as in the first experiment.

For the same reason as above, we only select the λ from $\frac{5}{\sqrt{3000}}$ to $\frac{18}{\sqrt{3000}}$ and we can see that in this range of λ, the perplexity drops significantly: in the case of $i = 5, 6$, the drop is nearly $30 - 35\%$!

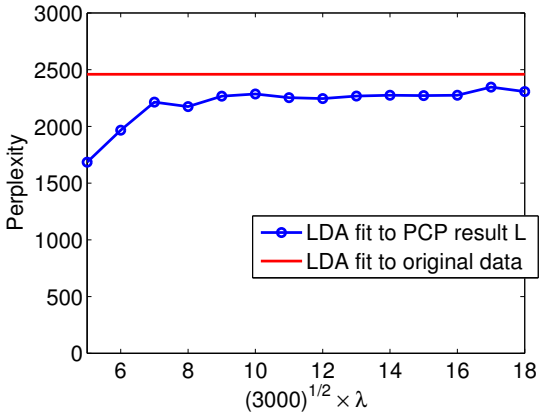
For $i = 5$, the matrix $L^{(5)}_{test}$ contains about only 76% of the vocabulary. This may have partially contributed to the drop in perplexity. Nevertheless, for $i = 6$, $L^{(6)}_{test}$ contains over 95% of the vocabulary, and the drop remains significant.

Even when λ increases significantly, for example for λ_{18}, the support of the sparse matrix $S^{(18)}_{test}$ is about $3,000$. This means that for each document with an average length of 500 words, we only treat 3 words as outliers but the perplexity still drops a lot.

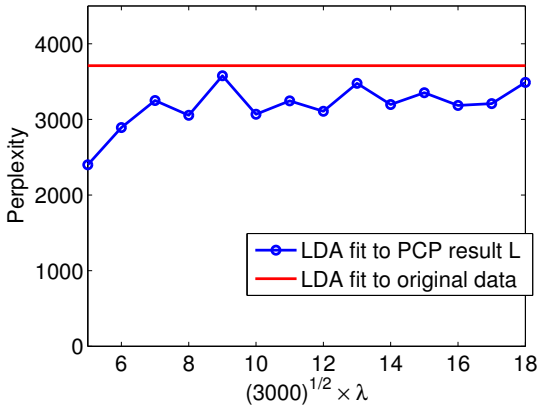
So we may conclude that the new method can effectively remove the outliers of a given corpus and the model learned after outliers being removed is much more generalizable.

3.3 Keywords Justification

In this section, we examine how so learned low-rank and sparse components encode useful information for document analysis. As we have discussed before, the low-rank matrix encodes the general topics that are contained in documents, whereas the sparse components are the document-specific terms. It is natural to ask that how well are the keywords



(a) Perplexity on Training Set



(b) Perplexity of Sequestered Test Set

Figure 4: Perplexity of topic model fit to Reuters data with and without PCP preprocessing. Left: Entire dataset as training. The blue line plots the perplexity of the fit to the low-rank component obtained via PCP, as a function of the weight factor λ in (8). The red line gives a baseline result without PCP preprocessing. Right: Testing result. The dataset is divided in half and both the training and test sets are decomposed via PCP. An LDA model is fit to the low-rank component for the training set. Plotted is the perplexity of the low-rank component of the test set with respect to this fitted model.

computed by PCP agree with human's intuition. It is, however, not easy to verify this due to the lack of well-labeled data. Nevertheless, in the experiment below, we examine the proportion that the keywords extracted by PCP are also the words contained in the title of each document. This is a reasonable test, since for most articles, especially formal ones, the titles are carefully chosen to represent the most informative and document-specific part in a corpus. We therefore consider the titles as the human label for the keywords of documents.

However, since title words are usually somewhat unique to each document, many of them will not fall in the list of 3,000 most frequent words in the corpus. So we have to resample the corpus to ensure most title words are included in the list of words of interest. By combining the title words with the most frequent words in the 1,000 documents in the corpus, we obtain an extended new data matrix D of size $3,322 \times 1,000$. We use another matrix X of the same size to denote the matrix of title words: x_{ij} is the number of times the ith word appears in the title of document j.

As in the previous section, we apply PCP to decompose the data matrix D into a low-rank part L and a sparse part S. We again choose 20 different λ's from $\frac{1}{\sqrt{3322}}$ to $\frac{20}{\sqrt{3322}}$ with step $\frac{1}{\sqrt{3322}}$. Then for each choice of λ_i, we get a different decomposition: $D = L^{(i)} + S^{(i)}$, for $1 \leq i \leq 20$.

We first investigate the relationships between the title words and the words that appear in the sparse matrix S. We compute the *correlation* between the title matrix X and each of the sparse matrix $S^{(i)}, i = 1, \ldots, 20$ in terms of two measures. We first normalize the sum of each column of $S^{(i)}$ to one and view it as a probability distribution:

1. The first measure, denoted as $t_1(i)$, is the fraction of entries in $S^{(i)}$ that are title words.

2. The second measure, denoted as $t_2(i)$, is the average of the sum of title entry values of each column of $S^{(i)}$. It can be viewed as the probability of any title word appearing in $S^{(i)}$.

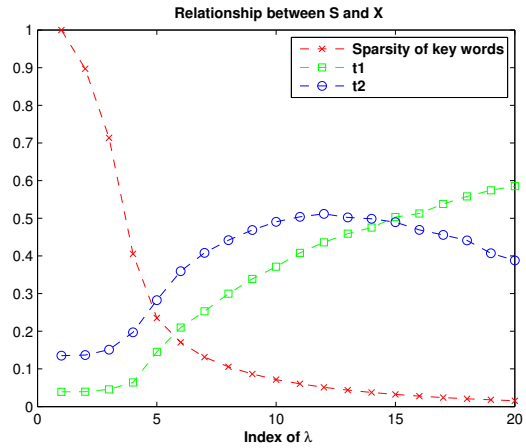


Figure 5: Keywords justification by comparing with Title Words. Two correlation measures t_1 and t_2 between the sparse matrix $S^{(i)}$ and the title matrix X.

We plot these measures as a function of i in Figure 5. To give a sense of how sparse the keywords detected by $S^{(i)}$ are in the corpus, we have also plotted the fraction of word counts given by $S^{(i)}$ w.r.t. the total number of words in the corpus.

As we see from the plot 5, the result is rather striking: The value of $t_2(i)$ reaches its peak at $i = 12$ or $\lambda = \frac{12}{\sqrt{3322}}$. The fraction of words detected by $S^{(12)}$ as keywords is about 5%. But out of the 5% keywords, more than 50% of them are title words (see Figure 1 for an example), even though we never told the algorithm which words belong to the titles! Note

from the same example that it is not necessarily true that title words are the most important words of a document – there may well be keywords that do not appear in the title. Such words will get detected by S as well. Also, it is interesting to see in the plot 5 that $t_1(i)$ is monotonic in i which suggests that as λ increases and S gets sparser, the percentage of title words left in the keyword list increases.

PCP decomposition actually provides us more information about the original dataset, *i.e.* background information versus document-specific information (or, geometrically, a proper subspace versus outliers). We have performed some preliminary experiments on information retrieval using the two parts in a cooperative manner. Specifically, we found that a better mAP can be obtained by weighting the low-rank terms and sparse terms separately, compared with traditional TF·IDF strategy. We leave a comprehensive comparisons with state-of-the-art and the possible ad-hoc strategies with *learning to rank* (see,*e.g.* [12, 13]) for future work.

4. DISCUSSION

In this paper, we have argued that sparse and low-rank models may be more relevant for text data analysis than more traditional topic models. We have further demonstrated that Principal Component Pursuit is an effective tool for decomposing the given data into sparse and low-rank components, and suggested that this can indeed improve retrieval performance. However, as a new tool, its full capabilities are still far from well-understood. Although we have focused on the retrieval task, we believe that this approach might be useful for a broad family of problems, including text classification, information retrieval, and document summarization, to name a few. Moreover, the scale could be extended to several order of magnitudes larger by the rapidly advancements in parallel and distributed computing.

5. REFERENCES

[1] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, O. Chapelle, and K. Q. Weinberger. Supervised semantic indexing. In *CIKM*, pages 187–196, 2009.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[3] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *preprint*, 2009.

[4] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *preprint*, 2009.

[5] C. Chemudugunta, P. Smyth, and M. Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. In *NIPS*, pages 241–248, 2006.

[6] C. H. Q. Ding. A similarity-based probability model for latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 58–65, 1999.

[7] S. T. Dumais. An interdisciplinary perspective on information retrieval. In *SIGIR*, pages 1–2, 2009.

[8] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI*, pages 281–286, 1988.

[9] A. Gelbukh and G. Sidorov. Zipf and heaps laws' coefficients depend on language. In *Conference on Intelligent Text Processing and Computational Linguistics*, pages 332–335, 2004.

[10] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

[11] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.*, 42(1-2):177–196, 2001.

[12] J.-W. Kuo, P.-J. Cheng, and H.-M. Wang. Learning to rank from bayesian decision inference. In *CIKM*, pages 827–836, 2009.

[13] T.-Y. Liu. Learning to rank for information retrieval. In *Foundations and Trends in Information Retrieval*, pages 225–331, 2009.

[14] J. Miao and A. Ben-Israel. On principal angles between subspaces in $\mathbb{R}^n$. *Lin. Algeb. and its Appl.*, 171:81–98, 1992.

[15] M. Mitra and B. B. Chaudhuri. Information retrieval from documents: A survey. *Inf. Retr.*, 2(2/3):141–163, 2000.

[16] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: a probabilistic analysis. In *Proceedings of the seventeenth ACM symposium on Principles of database systems*, pages 159–168, 1998.

[17] M. Steyvers and T. Griffiths. Probabilistic topic models. *Latent Semantic Analysis: A Road to Meaning*, 2007.

[18] X. Wei and B. W. Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, 2006.

[19] X. Yuan and J. Yang. Sparse and low-rank matrix decomposition via alternating direction methods. *preprint*, 2009.

[20] L. W. Z. Lin, M. Chen and Y. Ma. The augmentd lagrange multiplier method for exact recovery of a corrupted low-rank matries. *Mathematical Programming*, 2009.