Statistical Pattern Recognition : Experiments with Latent Dirichlet Allocation

Soubhik Barari

December 14 2015

1 Introduction

In this report, we outline the process and results of using Latent Dirichlet Allocation for document analysis and classification using some experimental datasets. We evaluate the results and describe some high-level conclusions about the LDA model.

2 Latent Dirichlet Allocation

Popularized by Blei et. al. in 2003, the Latent Dirichlet Allocation (LDA) model provides a generative Bayesian framework for topic modeling in a corpus of text documents. The generative process is as follows:

1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.
   (b) Choose a word w_n from $p(w_n | z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

Figure 1: Excerpt from 'Latent Dirichlet Allocation' by Blei et. al. [1]

In our case, we may substitute step 1. with a user-defined parameter N. However, note that at a high level, the LDA generative process is quite literally 'shaped' by the hyperparameters α and β which are shape parameters to a Dirichlet distribution.

We may then perform posterior inference on a given corpus of documents using the technique of collapsed Gibbs sampling. Other methods include variational inference and expectation-maximization.

Figure 2 describes the particular computational details for Gibbs sampling.

Algorithm 1 Collapsed Gibbs sampler for LDA

Require: Number of topics K, Dirichlet parameter for topic distribution α, Dirichlet parameter for word distribution β, number of iterations to run sampler N_{iters}, set of word indices $\{w(n)\}$, set of document indices $\{d(n)\}$, and initial set of topic indices $\{z(n)\}$, where $n = 1 \dots N_{words}$

Generate a random permutation $\pi(n)$ of the set $\{1, 2, \dots, N_{words}\}$
Initialize a $D \times K$ matrix of topic counts per document C_d, where D is the number of documents
Initialize a $K \times V$ matrix of word counts per topic C_t, where V is the number of words in the vocabulary
Initialize a $1 \times K$ array of probabalities P (to zero)
for $i = 1$ to N_{iters} **do**
  for $n = 1$ to N_{words} **do**
    $word \leftarrow w(\pi(n))$
    $topic \leftarrow z(\pi(n))$
    $doc \leftarrow d(\pi(n))$
    $C_d(doc, topic) \leftarrow C_d(doc, topic) - 1$
    $C_t(topic, word) \leftarrow C_t(topic, word) - 1$
    for $k = 1$ to K **do**
      $P(k) = \frac{C_t(k,word)+\beta}{V\beta+\sum_j C_t(k,j)} \frac{C_d(doc,k)+\alpha}{K\alpha+\sum_l C_d(doc,l)}$
    end for
    $topic \leftarrow$ sample from P
    $z(\pi(n)) \leftarrow topic$
    $C_d(doc, topic) \leftarrow C_d(doc, topic) + 1$
    $C_t(topic, word) \leftarrow C_t(topic, word) + 1$
  end for
end for
return $\{z(n)\}, C_d, C_t$

Figure 2: Algorithm for collapsed Gibbs Sampling for LDA parameter estimation (Khardon, Sheth)

The results of interest in our experiments are *document-topic* frequencies as well as *topic-word* frequencies.

3 Support Vector Machine

In the second part of our project, we are interested in binary classification of documents - using *different* feature representations. To perform classification, we use the popular Support Vector Machine (SVM) algorithm.

The SVM algorithm is an 'augmented' form of the two-class classification using the linear model

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$$

where $\phi(\mathbf{x})$ denotes the feature space transformation of data point $\mathbf{x}$, b is an explicit bias parameter and $y(\mathbf{x})$ is the prediction of class of point $\mathbf{x}$ which is determined by sign$(y(\mathbf{x}))$. Our objective is to find a hyperplane $\mathbf{w}$ that best separates data of the two classes, while maintaining correct classification of our known dataset. The optimization objective with constraints can formulated as:

$$\min \frac{1}{2} \|\mathbf{w}\|^2 \text{ s.t. } t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1$$

where t_n corresponds to the correct class label for data point x_n in our training dataset.

In the SVM model, we introduce slack variables to enforce the notion of a margin between the two classes. We introduce two different types of slack variables, ξ and $\hat{\xi}$, for slack outside the margin of class 1 and -1 respectively. The objective then becomes:

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^{N} (\xi_n + \hat{\xi}_n)$$

with the constraints:

$$t_n \leq y(\mathbf{x}_n) + \xi_n t_n \geq y(\mathbf{x}_n) - \hat{\xi_n}$$

We can then solve this optimization problem by considering the primal Lagrangian form and then solving the (easier) dual form.

The prime advantage of the SVM model is that it is a *kernel method*, which allows us to specify a 'method' for performing inner products $\phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m)$ as $k(\mathbf{x}_n, \mathbf{x}_m)$ from a variety of known kernel methods.

4 Results

We created an LDA collapsed Gibbs sampler from scratch using the aforementioned method. To evaluate our dataset, we used a subset of 100 documents from the *20 newsgroup* dataset, a collection of $20,000$ documents partitioned evenly across 20 news groups. This collection is a popular dataset in experiments with text analysis in machine learning.

4.1 LDA topic modeling

We tuned our model to detect $K = 20$ topics over 100 iterations. As 'default' hyperparameters, we set $\alpha = \frac{50}{K} = 2.5$ and $\beta = 0.01$. Table 1 reports some of the most frequent words of the subsequent 20 topics. Based on our own human judgment, it appears that topics are either related to 'sports' or 'computer hardware'.

4.2 SVM document classification

Using the support vector machine algorithm for binary classification, we compare the classification of *20 newsgroup* documents using both *topic frequency* and *bag-of-words* as our feature vectors for our documents. Figure 3 reports the learning curves for SVM document classification from an initial set of topics.

Topic	keywords
1	*shots, period, saves, stars, Chicago*
2	*game, great, rangers, Andrew, York*
3	*team, hockey, think, game, didn't*
4	*best, goal, third, powerplay, made*
5	*speed, system, memory, computer, much*
6	*know, card, games, monitor, apple*
7	*SCSI, drives, data, more, controller*

Table 1: LDA *topic-word* frequencies for $K = 20$ on *20 newsgroups*. We report the most 'human interpretable' topics on an intial run of our Gibbs sampler using default parameters.

The different costs correspond to different tunings of the constant C in the SVM optimization problem described in the section above.

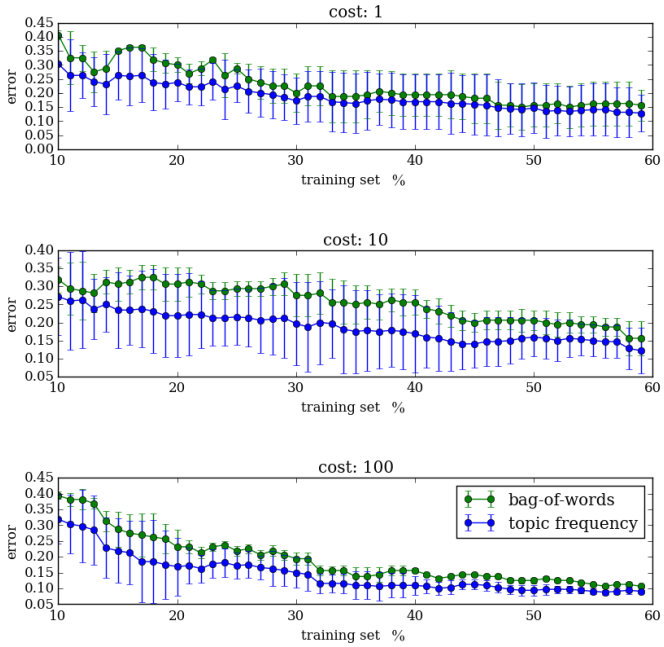

Figure 3: SVM prediction learning curves comparing different linear kernel costs of against different training set sizes. We plot the error rate of both *topic frequency* and *bag-of-words* feature vectors.

Visually, we notice that both $C = 1$ and $C = 100$ - more 'extreme' costs - generate learning curves with lower error variance for both feature representa-

tions. Particularly for $C = 100$, we achieve very low variance when fitting the SVM to the full training dataset.

This is because cost represents the idea of 'how much we want to avoid misclassifying datasets'. Smaller values of C will result in a larger margin hyperplane that is more lenient in allowing data points on the 'wrong' side. However, larger values of C will be much stricter and fit a smaller margin hyperplane to the training data. Since, we are training and testing on documents from the same corpus, the threat of overfitting is likely to be less potent - textual topic patterns (and linguistic patterns in general) are more consistent in a document corpus such as a collection of news articles.

Note that across all cost values, $\{1, 10, 100\}$, we see that the *topic frequency* feature representation outperforms the *bag-of-words* representation. This may be because we discovered topics in our LDA inference that produced particularly linearly separable feature vectors, or it may be a general benefit that we may reap using topic modeling.

Running our SVM prediction scheme on different costs and training splits again - this time on a *different* set of topics altogether from another run of Gibbs sampling - we get the results reported in Figure 4.

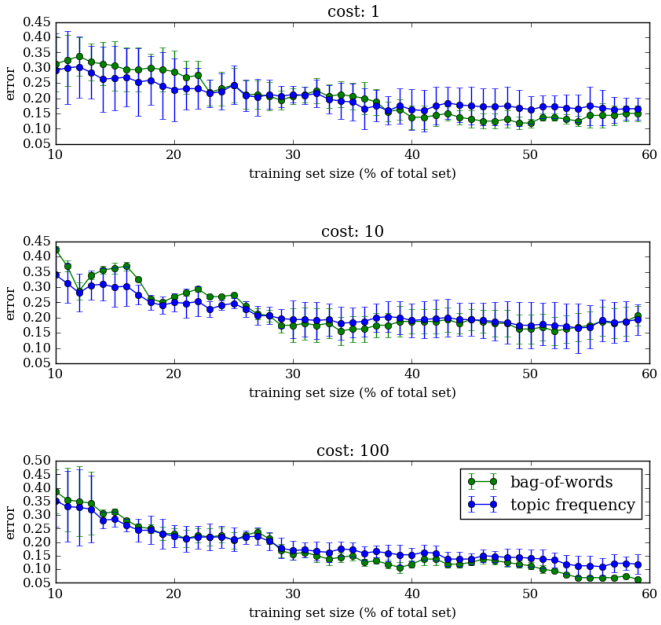

Figure 4: SVM prediction learning curves using another topic set.

From this second set of results, we cannot definitively say that the *topic frequency* feature representation outperforms the *bag-of-words* in producing a linearly sep-

arable feature set for a document in a corpus. There is further need for cross-validation in order to establish one representation as the better predictor in this particular dataset. However, note the tradeoffs between the two repreentations for a single document:

	feature cost	**dimensionality**	**classification cost**
bag-of-words	low	high	high
topic frequency	high	low	low

It is clear that performing the LDA algorithm is more computationally expensive than simply collapsing word documents into *bag-of-words* vectors. However, the SVM is left to deal with the entire dimensionality of the document words, so the cost is incurred in the classification stage. The Latent Dirichlet Algorithm, beyond just creating a human interpretable summary of a document corpus, provides an effective feature extraction and dimensionality reduction method for a text corpus - incurring little performance loss with enough data, as demonstrated in Figure 3 and Figure 4.

5 Conclusion

In this report, we experimented with a pipeline of topic modeling and document classification using the LDA and SVM algorithms respectively. Our results find that in document classification, with enough training data there is little difference in SVM classification, when our SVM model is tuned to the right cost (in this case, a stricter cost). We conclude that LDA is an effective dimensionality reduction tool, and based on Table 1, a friendly method for text summarization.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.