Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Local Context Sparse Coding

Seungyeon Kim* and **Joonseok Lee*** and **Guy Lebanon†** and **Haesun Park***
*College of Computing, Georgia Institute of Technology, Atlanta, GA, USA
†Amazon, Seattle, WA, USA
{seungyeon.kim, jlee716}@gatech.edu, glebanon@gmail.com, hpark@cc.gatech.edu

Abstract

The n-gram model has been widely used to capture the local ordering of words, yet its exploding feature space often causes an estimation issue. This paper presents local context sparse coding (LCSC), a non-probabilistic topic model that effectively handles large feature spaces using sparse coding. In addition, it introduces a new concept of locality, local contexts, which provides a representation that can generate locally coherent topics and document representations. Our model efficiently finds topics and representations by applying greedy coordinate descent updates. The model is useful for discovering local topics and the semantic flow of a document, as well as constructing predictive models.

1 Introduction

Learning a representation that reflects word locality is important in a wide variety of text processing applications such as text categorization, information retrieval, or language model generation. The n-gram model, for example, is popular because of its simplicity and efficiency, which interprets a document as a collection of word sub-sequences. Specifically, it models a word given the previous $n-1$ words: $p(w_i|w_{i-1},\ldots,w_{i-n+1})$. The larger n is, the longer the contexts that the model can capture. A related approach is to model a symmetric window around a word $p(w_i|w_{i+1},w_{i-1},w_{i+2},w_{i-2},\ldots)$, as is done for example by Mikolov et al. (2013).

Lebanon, Mao, and Dillon (2007) extended local dependencies by applying different weights at each position of a document and summing up the word presence near a particular location. Specifically, that approach, named "locally weighted bag-of-words" (LOWBOW), uses a smoothing kernel to generate a smooth curve in the probability simplex that represents the temporal progression of the document. LOWBOW allows examining much longer-range dependencies than n-gram models, and it also allows tying word patterns to specific document locations. The bandwidth of the smoothing kernel captures the tradeoff between estimation bias and estimation variance. Our approach extends their work, but is different as it decouples local probabilities from their positions and it uses sparse coding to compress the parameter space.

Document models such as the n-gram and LOWBOW suffer from intrinsic sparsity, an inevitable consequence of capturing dependencies in sequences over a large vocabulary. The larger the dependency range, the harder it is to estimate the dependencies due to increased estimation variance. Specifically, the number of possible combinations of n consecutive words grows exponentially, making the number of observations for each combination extremely sparse, eventually causing not only computational difficulties but also a high estimation error. As a result, in many cases where data is limited, n-gram models with low n perform better than n-gram models with high values of n.

Neural probabilistic language models such as Bengio et al. (2006) are an attempt to handle this issue. They capture long term relations over a large vocabulary by using a parametric model that compresses the parameter space. Since the model estimates a compressed parameter vector rather than the exponentially growing n-gram counts, it is an effective way of capturing word dependencies that n-gram models cannot. On the other hand, probabilistic topic models such as Blei, Ng, and Jordan (2003) and matrix decomposition models (Deerwester et al. 1990; Lee and Seung 1999; Zhu and Xing 2011) estimate a compressed representation of the vocabulary, usually termed latent space or topics. Unlike the neural language model, these models are usually based on the bag-of-words representation or bigram features (Wallach 2006), limiting their potential to capture sequential word dependencies (though some recent extensions generalize topic models to sequential models - see Section 2).

By efficiently estimating sparse and compact representations of local dependencies, our model extends the work of Lebanon, Mao, and Dillon (2007) and Zhu and Xing (2011). We first define the notion of a *local context*, which is a conditional word probability given the word's location in the document. Similar to Lebanon, Mao, and Dillon (2007), we use a smoothing kernel to estimate the local context. Each kernel bandwidth examines a unique range of local resolutions. As noted earlier, because of the huge number of local contexts in our model, we apply a sparse-coding formulation to compress the space.

Our model has several benefits. First, by introducing rich local dependencies, it can generate highly discriminating

features. Second, it produces a sparse and compact representation of a document. Third, since it also models word proximities, it can be used to generate locally coherent topics that will be a useful tool for analyzing the topical flow of a document.

2 Related Work

Recent studies on modeling document locality have focused on variations of the n-gram model. For example, Mikolov et al. (2013) connects a variant of the n-gram model with a neural probabilistic language model. A different approach to extending n-grams was taken by Lebanon, Mao, and Dillon (2007), who used kernel smoothing on length-normalized documents. Similar ideas were also explored in Mao, Dillon, and Lebanon (2007); Lebanon and Zhao (2008) and Lebanon, Zhao, and Zhao (2010). Our model conveys a new concept of locality by combining n-gram and kernel smoothing.

Text segmentation and parsing studies also focus on local document features. For example, Chen et al. (2009) introduced semantic segments using a hierarchical topic model. Unlike our paper that focuses on spatial segments, these studies focus on semantic segments resulting in a semantic locality concept.

Our method adopts a topic-modeling approach to compress a large feature space, an unavoidable outcome of long-range dependencies. Our approach extends Sparse Topical Coding (STC), a non-probabilistic approach, that was shown to have state-of-the-art accuracy as well as relatively fast training time. A detailed comparison between STC and probabilistic topic models (Blei 2012) appears in Zhu and Xing (2011). Unlike standard matrix factorization methods such as non-negative matrix factorization (Lee and Seung 1999), STC uses sparsity constraint explicitly. Our model differs from STC in two ways: (i) it uses local contexts $p(w|t)$ instead of single-word observations $p(w)$ which leads to a different loss function, and (ii) instead of using pathwise coordinate descent, our model employs a new update rule based on greedy coordinate descent.

Temporal topic models (Blei and Lafferty 2006; Wang, Blei, and Heckerman 2009) are extensions of basic LDA that model sequential word appearances. They are similar to our model in that both approaches produce topics that vary across different document locations. Our approach differs from temporal topic models in that the sequential transitions are based on specific locations in the document, as in locally weighted bag-of-words, and our models feature the idea of kernel smoothing from non-parametric statistics.

3 Local Context

Most document and topic modeling studies use sequential features such as unigram or n-gram to model documents. Instead, Lebanon, Mao, and Dillon (2007) modeled a document as a joint distribution of words and their locations $p(w,t)$, where w is a word and t is the location. The joint distribution $p(w,t)$, estimated by kernel density estimation, models the probability that a word will occur at a specific index within the document. Although the approach is useful for modeling document progression, it cannot model the relative positioning of words. On the other hand, $p(w|t)$ can model the relative positioning of words.

A *local context* is the distribution of words that occurred near a specific document position: $p(w|t)$. We denote it by $\phi(t)$:

$$\phi : \mathbb{N} \to \mathbb{R}^{|V|} \quad \text{where } |V| \text{ is the size of vocabulary.}$$

Given a length L document $x = [w_1, \ldots, w_L]$ and a position i, we can estimate the local context $\phi(i)$ using a smoothing kernel $k(i,j)$ that is a real valued normalized function that is monotonic decreasing in $|i-j|$. Intuitively, the kernel defines the locality that we are interested in.

$$\phi(i) = \left[\phi_1(i), \ldots, \phi_{|V|}(i)\right]^\top$$
$$\phi_v(i) = \sum_{j=1}^{L} k(i,j)\mathbf{1}_{\{w_j=v\}}.$$
$$\text{s.t. } \sum_v \phi_v(i) = 1, \quad \forall_v\, \phi_v(i) \geq 0$$

3.1 Choices of $k(i,j)$

There are several standard choices of a smoothing kernel $k(i,j) = g(i-j)$. We follow Lebanon, Mao, and Dillon (2007) and use the Gaussian kernel, which is a normalized Gaussian density. However, for illustration purposes, we use below the constant kernel (for a support of 3 words)

$$k'(i,j) = \begin{cases} 1/3 & \text{if } |i-j| \leq 1 \\ 0 & \text{else.} \end{cases}$$

This kernel measures the existence of a word in the window $\{w_{i-1}, w_i, w_{i+1}\}$. It differs from the trigram representation in that it ignores the ordering within the window.

Non-constant kernels such as Gaussian kernels allow emphasizing words closer to the center of the window while discounting more remote locations.

3.2 Comparision with n-gram Models

The n-gram model and its variations fundamentally differ from our model since they use a joint distribution of consecutive words, $p(w_i, \ldots, w_{i-n+1})$, instead of a conditional distribution between words and locations, $p(w|t)$. The size of the event space of an n-gram model expands exponentially when either its vocabulary or the size of window (n) grows. By contrast, the event space of our model is invariant of the window size (or the kernel bandwidth) and only linear in vocabulary size. In practice, the n-gram model performs poorly when both the vocabulary and n are large. See Section 5.3 for empirical results.

4 Local Context Sparse Coding (LCSC)

We now consider the bag of local contexts of a document, $\Phi = \{\phi(i) : i = 1, \ldots, L\}$ (where L is the length of the document). Since direct estimation of bag-of-local-contexts statistics is intractable, we approximate each $\phi(i)$ using a

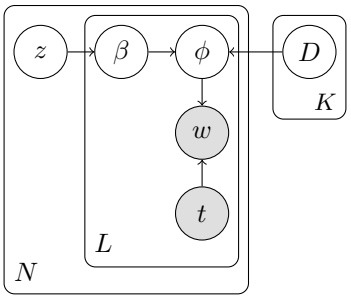

Figure 1: Graphical model of local context sparse coding. z denotes a document representation, and ϕ denotes a local context in a document of length L. D is a shared dictionary (topics), and β is a latent representation of a corresponding local context using D. See Section 4.1 for details.

linear combination of a handful (sparse) of codes in a dictionary of K codes (or *topics*).

$$\phi(i) \approx D\beta(i) \text{ where } D \in \mathbb{R}^{|V|\times K}, \beta(i) \in \mathbb{R}^K$$
$$\text{s.t. } \beta(i) \text{ is sparse, } D \geq 0, \ \forall_i \textstyle\sum_j D_{ij} = 1$$

Note in particular that the dictionary can be shared across multiple documents, and as a result the β that corresponds to different Φ (documents) are comparable.

We measure the approximation quality using the sum of squared distances between each $\phi(i)$ and $D\beta(i)$ and add a L_1 penalty on $\beta(i)$ to enforce sparsity. This standard practice is equivalent to maximizing the penalized likelihood of the model under a Gaussian distribution (regression) with a Laplace prior $p(\beta) \propto e^{-\lambda|\beta|}$ corresponding to the L_1 penalty. Thus, we get the following objective function for learning the dictionary D and the β parameters

$$\sum_{i=1}^{L} \|\phi(i) - D\beta(i)\|_2^2 + \lambda\|\beta(i)\|_1 \tag{1}$$

subject to the constraints of $D \geq 0$ and $\forall_i \sum_j D_{ij} = 1$. When we have multiple documents, we combine multiple squared error terms where the D matrix is shared and β parameters correspond to different documents as in (5).

Alternatively, we can use non-squared error loss functions as in Lee et al. (2009). In our experiments, we used the Hellinger distance $\|\sqrt{\phi(i)} - \sqrt{D\beta(i)}\|_2^2$, which performed the best. See Lebanon (2005a; 2005b), and Dillon et al. (2007) for additional examples of using Hellinger distances in text modeling and interpreting it in terms of information geometry.

We assume that the topic assignment parameters for a specific document are normally distributed $\beta(i)|z \sim \mathcal{N}(z, \rho^{-1}\,\mathrm{I})$ and consider its mean z as a document-specific parameter, or a *document representation*. This leads to the above objective function

$$\sum_{i=1}^{L} \rho\|\beta(i) - z\|_2^2 + \|\phi(i) - D\beta(i)\|_2^2 + \lambda\|\beta(i)\|_1 \tag{2}$$

subject to $D \geq 0$ and $\forall_i \sum_j D_{ij} = 1$. The equations above assume a single document. In the case of multiple documents, we sum over them as described in Section 4.2. In this case, D is shared across documents and β and z are document specific.

4.1 Comparison with Probabilistic Topic Models

The proposed method forms a graphical model as described in Figure 1, with the details appearing below. We follow some of the ideas in Zhu and Xing (2011) and note the caveat that the normalization in our model may not be consistent with the true distribution generating the data due to the fact that the parameters lie in a restricted domain (see comment below).

1. The local probability of words (or a local context) follows a distribution centered on $D\beta$ where D contains topics shared across multiple documents and β contains a corresponding topic assignment. For example, assuming a Gaussian distribution, we have:
$$\phi = p(w|t) \sim \mathcal{N}(D\beta, \sigma_\phi\,\mathrm{I}). \tag{3}$$
2. Topic assignments parameters $\{\beta(i) : i = 1, \ldots, L\}$ that correspond to a specific document follow a normal distribution centered on z with a Laplace prior.
$$\beta|z \sim \mathcal{N}(z, \rho^{-1}\,\mathrm{I}), \quad \beta \sim \text{Laplace}(0, \lambda^{-1}) \tag{4}$$

Traditional probabilistic topic models differ from our model primarily in two ways. First, instead of a single word observation $p(w)$, we model word locality through the distribution $p(w|t)$. Second, we do not directly compute the normalization terms of each probabilistic distribution. We only compute the numerator, for example $\|\beta(i) - z\|_2^2$, which is consistent with a Gaussian distribution but ignores the fact that β cannot achieve all values in a Euclidean space. This relaxation reduces the overall computation when compared to standard probabilistic topic models.

4.2 Estimation

The training procedure of our model is similar to the one of standard sparse coding models. Assuming we have multiple documents $X = [x^{(1)}, \ldots, x^{(N)}]$, we minimize the aggregated loss function of (2),

$$\min_{\beta,z,D} \ell = \min_{\beta,z,D} \sum_{n=1}^{N} \sum_{i=1}^{L^{(n)}} \Big[\rho\|\beta^{(n)}(i) - z^{(n)}\|_2^2 + \|\phi^{(n)}(i) - D\beta^{(n)}(i)\|_2^2 + \lambda\|\beta(i)\|_1 \Big] \tag{5}$$

subject to the following constraints on the shared dictionary D: $D \geq 0$ and $\forall_i \sum_j D_{ij} = 1$. It is a biconvex problem that can be iteratively solved for β, z and D. We additionally include non-negativity constraint on β for better interpretability, similar to Zhu and Xing (2011).

Solving for β and z By repeatedly optimizing each dimensions of β (coordinate descent), the lasso problem can be solved in closed form and have a unique solution under the non-negativity constraint. Specifically, using the shorthand notation $\beta^{(n)}(i) \to \beta$, $z^{(n)} \to z$, $\phi^{(n)}(i) \to \phi$, minimizing a single component of $\beta^{(n)}(i)$ gives the following:

$$\min_{\beta_j} \sum_{k=1}^{K} \rho(\beta_k - z_k)^2 + \sum_{v=1}^{|V|} \left(\phi_v - \sum_{k=1}^{K} D_{vk}\beta_k \right)^2 + \sum_{k=1}^{K} \lambda|\beta_k|$$

$$= \min_{\beta_j} \Bigg[\underbrace{\left(\rho + \|D_{:j}\|_2^2\right)}_{a} \beta_j^2 - 2 \underbrace{\left(\rho z_j + \sum_{v=1}^{|V|} D_{vj} \left(\phi_v - \sum_{k \neq j} D_{vk}\beta_k \right) \right)}_{b} \beta_j + \lambda|\beta_j| \Bigg].$$

The corresponding optimal solution is

$$\beta_j = \frac{1}{a} \min\left(0, b - \frac{\lambda}{2}\right). \tag{6}$$

The corresponding document representation $z^{(n)}$ also can be solved in closed form since we are minimizing L_2 distances between $z^{(n)}$ and $\beta^{(n)}(1), \ldots, \beta^{(n)}(L^{(n)})$.

$$z^{(n)} = \frac{1}{L^{(n)}} \sum_i \beta^{(n)}(i). \tag{7}$$

We would normally iterate the dimensions of β in a sequential order ($j = 1, 2, \ldots, K$) until convergence, which is called pathwise coordinate descent as was done in the training of STC (Zhu and Xing 2011). Greedy coordinate descent (Li and Osher 2009), however, updates one dimension at a time by choosing the dimension that reduces the loss the most ($\Delta\ell$). This results in faster training than pathwise method with the same accuracy level. See Li and Osher (2009) for detailed discussion.

By applying greedy coordinate descent and exploiting the factorization of the loss function, we developed an efficient algorithm for β and z (see Algorithm 1). Since greedy coordinate descent ensures the difference between β^{t+1} and β^t is exactly $\beta_j^{t+1} - \beta_j^t$ (j is the updated dimension), b and z can be updated efficiently using the previous values of those. In addition, β and z of a document are independent from those of other documents, and $\{\beta(i) : i = 1, \ldots, L\}$ in a single document only shares z during the update, which allows parallelization. Note that we approximate the loss decrease $\Delta\ell$ by $|\tilde{\beta}(i) - \beta^t(i)|$ (see Algorithm 1 for details.)

Solving for D Projected gradient descent method efficiently optimizes the dictionary D under the simplex constraint ($D \geq 0,\ \forall_i \sum_j D_{ij} = 1$).

$$\min_D \ell(D) = \min_D \sum_{n=1}^{N} \sum_{i=1}^{L^{(n)}} \|\phi^{(n)}(i) - D\beta^{(n)}(i)\|_2^2 \tag{8}$$

$$\nabla\ell(D) = -2 \sum_{n=1}^{N} \sum_{i=1}^{L^{(n)}} \left(\phi^{(n)}(i) - D\beta^{(n)}(i)\right) \beta^{(n)}(i)^\top \tag{9}$$

Specifically, we take a gradient step based on the gradient above and then project back to the simplex using a simplex projection Π.

$$D^{t+1} = \Pi(D^t - \eta_t \nabla). \tag{10}$$

Algorithm 1 Greedy coordinate descent for β and z

Input: local contexts of $x^{(1)}, \ldots, x^{(N)}$ and D
for all $x \in \{x^{(1)}, \ldots, x^{(N)}\}$ **do in parallel**
  $\Phi = [\phi(1), \ldots, \phi(L)]$ in x
  $[b(1), \ldots, b(L)] = D^\top \Phi$
  $z = \frac{1}{L} \sum_i b(i)$
  while $\sum_i |\beta(i)^{t+1} - \beta(i)^t| > \epsilon$ **do**
    $z^{t+1} = z^t$
    for all $i \in \{1, \ldots, L^{(n)}\}$ **do in parallel**
      $\tilde{\beta}(i) = \frac{1}{a} \min(0, b(i) - \lambda/2)$
      $j = \arg\max_k |\tilde{\beta}(i)_k - \beta(i)_k^t|$

      $\beta(i)^{t+1} = \begin{cases} \tilde{\beta}(i)_j & \text{at } j\text{th dimension} \\ \beta(i)^t & \text{else} \end{cases}$
      $z_j^{t+1} = z_j^t + (\beta_j^{t+1} - \beta_j^t)/L$
      $b(i)^{t+1} = b(i)^t - (\beta(i)_j^{t+1} - \beta(i)_j^t)(D^\top D)e_j$

      # wait for others to finish updating z
      $b(i)_j^{t+1} = b(i)_j^t + \rho(z_j^{t+1} - z_j^t)$
    end for
  end while
end for
Output: $z^{(1)}, \ldots, z^{(N)}$ and all β for all local contexts.

The projection Π can be computed efficiently, see for example Duchi et al. (2008) for details. We estimate the step size η by a line search that minimizes the dictionary related loss $\min_\eta \sum \|\phi - D^{t+1}\beta\|_2^2$.

5 Experiments

5.1 Illustrating Example

We illustrate the proposed method (LCSC) using a synthetic example of four documents with two different types of word locality: $\{a, b\}$ vs $\{a, c\}$.

$$x_1 = [a, b, a, b, a, b, c, c, c],\ x_2 = [b, a, b, a, b, a, c, c, c]$$
$$x_3 = [a, c, a, c, a, c, b, b, b],\ x_4 = [c, a, c, a, c, a, b, b, b]$$

While a and b accompany together in x_1 and x_2, a and c are together in x_3 and x_4, resulting in the topics of x_1 and x_2 being different from the topics of x_3 and x_4.

Bag-of-words representation, a common feature for topic models, generates exactly the same representations $[3, 3, 3]$ or $[0.33, 0.33, 0.33]$ (normalized) for all documents. By contrast, the bigram model distinguishes all four documents although it strictly separates two locally similar pairs ($[a, b]$ and $[b, a]$) at the same time. Despite the fact that the strict separation might be a preferable choice, this will eventually lead to an explosion of the feature space (especially when trying to account for long-range dependencies). See Section 3.2 for detailed discussion.

Unlike n-gram models, LCSC easily captures two topics corresponding to two distinct types of locality. Figure 2 shows the result of LCSC in a simplex using a dictionary of size $K = 2$ (number of topics) and a Gaussian smoothing kernel with bandwidth of 0.7. The smoothing kernel

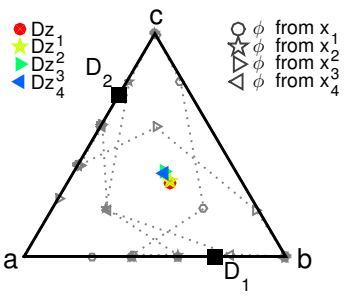

Figure 2: Result of LCSC on the synthetic example of Section 5.1 in a simplex, each corner of which represents the probability of one of the corresponding character. Filled shapes (Dz) denote document representations on the simplex; unfilled shapes (ϕ) are for local contexts of each document; filled squares are for two topics D_1, D_2. We see clear separation between $\{Dz_1, Dz_2\}$ vs $\{Dz_3, Dz_4\}$.

covers an effective width of about 5 words (weighted non-uniformly).

Figure 2 visualizes the characteristics of the dataset. First, two topics D_1 and D_2 capture two different types of locality. D_1 is located between a and b denoting the mixture topic of a and b; D_2 is located between a and c. Second, document representations on the simplex (Dz) form two separate groups. The first group consists of Dz_1 and Dz_2 and the second group consists of Dz_3 and Dz_4. The positions of the document representations discriminate documents by its local word distribution $p(w|t)$. Note that n-gram model cannot easily achieve this.

5.2 Local Topics

In contrast to the topics of traditional topic models, LCSC topics reflect the word locality. For instance, Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003) will fail to capture any meaningful topics on the synthetic example of Section 5.1 because all four documents have the same uniform word distribution. Unlike LDA, LCSC discovered two topics corresponding to two distinct types of locality in the previous section. In addition, as each local context contains its neighborhood information, LCSC eventually forms locally coherent topics, which are useful in practice since most text in general have locally coherent contents.

We compare LCSC with a well known topic-modeling technique, LDA, on a real world data: a Wikipedia article "Paris." We chose the article because it contains common knowledge and is well structured, albeit we do not use any structural information.

Figure 3 shows topic assignments at each position of the Paris article by LDA and LCSC (K=15 for both). The document progresses from left to right and each position corresponds to a word. The top figure (LDA) does not show any locally coherent structure, which is rather fragmented into pieces. In the bottom figure (LCSC), the topic assignments are locally coherent and illustrate the semantic flow of the document; it starts with the introduction of the city: general information (topic 1 on Table 1) and its reputation (topic 2), which are followed by several aspects of Paris: history (topic 3,4), exposition (topic 5), art (topic 6), and education (topic 7). In addition, top words of each topic are indeed highly indicative of each local subject (Table 1).

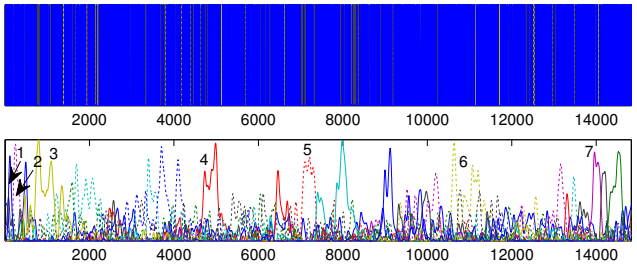

Figure 3: Topic assignments at each position of Wikipedia article "Paris" by LDA (top) and LCSC (bottom). The leftmost edge indicates the beginning of the document and the rightmost edge for the end. LCSC topics are more locally distributed than LDA. Numbers on the bottom figure indicate topic IDs; Table 1 has the detail of each topic.

1	mi km sq area population kilometres bois city north paris river climate arrondissements vincennes south
2	world fashion paris international high cent largest manufacturing business million europe region global
3	roman bc parisii century found seine bank romans lutetia ad left le site cit soldiers age excavations built
4	king national government july commune paris sans culottes city army guard palace festival revolution
5	exposition champs universal visitors eiffel tower mars meters held world palais million iii hosted place
6	theatre arrondissement des tel du located musee district ra including centre op paris place theatres lies
7	library paris arrondissement libraries le biblioth university public located sorbonne mitterrand ois fran

Table 1: Top words of selected topics using LCSC on a Wikipedia article "Paris." See text for details.

We also tried other types of documents that are not structurally written, such as novels ("The Metamorphosis" by Kafka, "The Last Leaf" by O. Henry), a speech ("I Have a Dream" by MLK), and an editorial (a Watergate article), and they all demonstrated an ability to learn locally coherent topics.

5.3 Classification

We examine in this section using features generated by LCSC in classification. We used a standard classifier, support vector machine[1], with different sets of features. Specifically, we used ν-SVM whose ν value was selected from 10 candidate values using cross-validation.

Our classification task was the standard 20 newsgroup[2] classification data with the official train-test split and standard preprocessing: lowercasing, stripping non-english characters, tokenizing sentences and words, Porter stemming, and removing rare features and stop words. The preprocessing resulted in 18846 documents, 20 classes, and vocabulary

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[2] http://qwone.com/~jason/20Newsgroups/

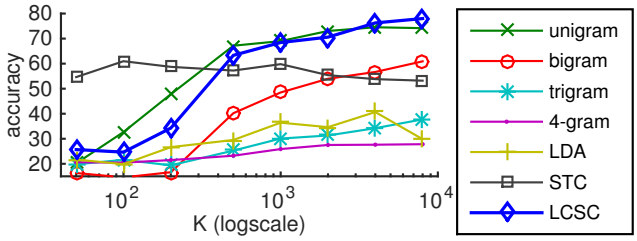

Figure 4: Test set classification accuracies with various sizes of dictionaries and methods

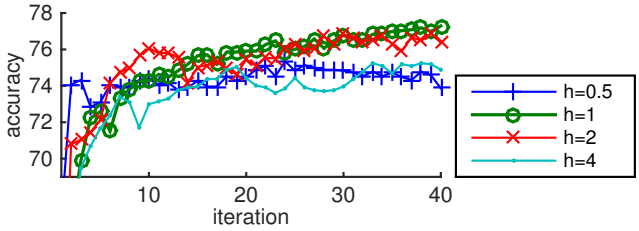

Figure 5: Test set classification accuracies of LCSC with various smoothing bandwidths

of size $|V| = 6328$. In the following two subsections, to examine the effect of parameters, we handle a subset of the dataset (5 classes, comp.*). In the last subsection, we evaluate overall performance on both the subset of the dataset and the whole dataset.

Effect of the Number of Topics (K) Figure 4 shows test set classification accuracies with various methods and sizes of dictionaries (from 50 to 8000). In the case of n-gram models, we selected the most frequent K features from the training set. For the other methods LDA[3], STC[4], and LCSC, we specify the size of a dictionary as a parameter. The bandwidth of LCSC was fixed to h=1, which covers about 7 words ($\pm 3h$). We tried a set of candidates for the remaining parameters and chose the best performing one (for example, $\lambda = \{10^{-4}, 10^{-2}, 10^{-1}, 0.5, 1\}$ for STC).

LCSC performs similar to unigram with small dictionaries, but it eventually achieves superior performance with a dictionary of sufficient size (from K=4000), that is, the performance of LCSC keeps improving even after $K > |V|$ (unigram model reaches maximum performance when $K < |V|$). STC performs well with relatively small dictionaries, but its maximum performance is not as good as other methods.

Figure 4 partially confirms Section 3.2. Bigram, trigram and 4-gram model do not perform well even with a large dictionary. It is because the number of features grows rapidly (bigram generates $23|V|$ features, trigram for $35|V|$, and 4-gram for $37|V|$) and thus will drastically lower the number of observations for each feature. On the contrary, even though LCSC covers approximately 7 neighboring words, it does not seem to suffer from sparsity and shows superior performance.

[3]FastLDA: http://www-users.cs.umn.edu/~shan/mmnb_code.html

[4]http://www.ml-thu.net/~jun/stc.shtml

	n-gram	LDA	STC	LCSC	MedSTC
comp.*	74.53	40.67	60.97	**78.01**	77.70
*	74.10	34.43	61.14	**80.76**	79.81

Table 2: Comparision of test set classification accuracy for various methods on 5 classes (comp.*) and full 20 classes (*) of 20 newsgroup dataset

Effect of Bandwidth (h) Figure 5 shows test set classification accuracies of LCSC with various bandwidths while other parameters are fixed (K=4000, ρ=10^{-4}, λ=10^{-2}). The best performance was obtained at h=1. Using narrower bandwidth (h=0.5) led to faster convergence to poor performance, which is caused by lack of variability of local features. Using broader bandwidth (h=4) slowed down the convergence and ruined the performance, which is attributed to including unnecessary local dependencies for this task. The diverse results of various bandwidths confirms that locality features makes a notable difference in classification performance.

Comparision of Overall Performance We finally compare the overall performance of LCSC with other methods including a local-dependency model, n-gram, and unsupervised topic models: LDA and STC. We additionally included a top performing *supervised* topic model, MedSTC (Zhu and Xing 2011). Note, however, that MedSTC uses auxiliary supervised information (labeled data) during its topic learning, and cannot be directly compared to our method. We tried various sets of parameters and choose the best performing one (K: [50,...,8000], λ, ρ: [10^{-4},...,10^{-1}]). For n-gram models, we tried n: [1,...,4] and chose the best.

LCSC outperforms all other competitors on the subset as well as the full set (Table 2). The performance gain with respect to n-gram models shows that modeling long-range dependencies can be beneficial in classification. The better performance of LCSC compared to other methods including MedSTC (significant at p-value: 0.002) is notable since MedSTC directly optimizes for its discriminative performance whereas LCSC is a purely unsupervised coding method.

6 Summary

This paper presents a non-probabilistic topic model for local word distributions. Our model employed kernel smoothing to capture sequential information, which granted a flexible and efficient way to handle a wide range of local information. Our sparse-coding formulation leads to efficient training procedures, and a sparse representation that is locally coherent and has stronger discrimination capacity.

Acknowledgments

This work was supported in part by NSF Grant CCF-1348152 and DARPA XDATA Grant FA8750-12-2-0309. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Bengio, Y.; Schwenk, H.; Senécal, J.; Morin, F.; and Gauvain, J. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*. Springer. 137–186.

Blei, D., and Lafferty, J. 2006. Dynamic topic models. In *Proc. of the International Conference on Machine Learning*.

Blei, D.; Ng, A.; and Jordan, M. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Blei, D. M. 2012. Probabilistic topic models. *Communications of the ACM* 55(4):77–84.

Chen, H.; Branavan, S.; Barzilay, R.; and Karger, D. 2009. Content modeling using latent permutations. *Journal of Artificial Intelligence Research* 36(1):129–163.

Deerwester, S.; Dumais, S.; Landauer, T.; Furnas, G.; and Harshman, R. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science* 41(6):391–407.

Dillon, J.; Mao, Y.; Lebanon, G.; and Zhang, J. 2007. Statistical translation, heat kernels, and expected distances. In *Uncertainty in Artificial Intelligence*, 93–100. AUAI Press.

Duchi, J.; Shalev-Shwartz, S.; Singer, Y.; and Chandra, T. 2008. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proc of International Conference on Machine Learning (ICML)*.

Lebanon, G., and Zhao, Y. 2008. Local likelihood modeling of the concept drift phenomenon. In *Proc. of the 25th International Conference on Machine Learning*.

Lebanon, G.; Mao, Y.; and Dillon, J. 2007. The locally weighted bag of words framework for documents. *Journal of Machine Learning Research* 8:2405–2441.

Lebanon, G.; Zhao, Y.; and Zhao, Y. 2010. Modeling temporal text streams using the local multinomial model. *Electronic Journal of Statistics* 4.

Lebanon, G. 2005a. Axiomatic geometry of conditional models. *IEEE Transactions on Information Theory* 51(4):1283–1294.

Lebanon, G. 2005b. Information geometry, the embedding principle, and document classification. In *Proc. of the 2nd International Symposium on Information Geometry and its Applications*, 101–108.

Lee, D., and Seung, H. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401:788–791.

Lee, H.; Raina, R.; Teichman, A.; and Ng, A. 2009. Exponential family sparse coding with application to self-taught learning. In *International Joint Conferences on Artificial Intelligence*.

Li, Y., and Osher, S. 2009. Coordinate descent optimization for l1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Probl. Imaging* 3(3):487–503.

Mao, Y.; Dillon, J.; and Lebanon, G. 2007. Sequential document visualization. *IEEE Transactions on Visualization and Computer Graphics* 13(6):1208–1215.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *Workshop at International Conference on Learning Representations*.

Wallach, H. 2006. Topic modeling: beyond bag-of-words. In *Proc. of the International Conference on Machine Learning*.

Wang, C.; Blei, D.; and Heckerman, D. 2009. Continuous time dynamic topic models. In *Proc. of Uncertainty in Artificial Intelligence*.

Zhu, J., and Xing, E. 2011. Sparse topical coding. *In Proc. of Uncertainty in Artificial Intelligence (UAI)*.