Multiscale Analysis of Document Corpora Based on Diffusion Models

Chang Wang and **Sridhar Mahadevan**
Computer Science Department
University of Massachusetts
Amherst, Massachusetts 01003
{chwang, mahadeva}@cs.umass.edu

Abstract

We introduce a nonparametric approach to multiscale analysis of document corpora using a hierarchical matrix analysis framework called diffusion wavelets. In contrast to eigenvector methods, diffusion wavelets construct multiscale basis functions. In this framework, a hierarchy is automatically constructed by an iterative series of *dilation* and *orthogonalization* steps beginning with an initial set of orthogonal basis functions, such as the unit-vector bases. Each set of basis functions at a given level is constructed from the bases at the lower level by dilation using the dyadic powers of a diffusion operator. A novel aspect of our work is that the diffusion analysis is conducted on the space of variables (words), instead of instances (documents). This approach can automatically and efficiently determine the number of levels of the topical hierarchy, as well as the topics at each level. Multiscale analysis of document corpora is achieved by using the projections of the documents onto the spaces spanned by basis functions at different levels. Further, when the input term-term matrix is a "local" diffusion operator, the algorithm runs in time approximately linear in the number of non-zero elements of the matrix. The approach is illustrated on various data sets including NIPS conference papers, 20 Newsgroups and TDT2 data.

1 Introduction

The problem of analyzing text corpora has emerged as one of the most active areas in data mining and machine learning. The goal here is to extract succinct descriptions of the members of a collection that enable efficient generalization and further processing. Many real-world corpora of text documents exhibit non-trivial semantic regularities at *multiple* levels, which cannot be easily discerned using "flat" methods, such as Latent Semantic Indexing (LSI) [Deerwester *et al.*, 1990]. For example, in the well-known NIPS conference paper data set, at the most abstract level, the set of all papers can be categorized into two main topics: machine learning and neuroscience. At the next level, the papers can be categorized into a number of areas, such as dimensionality reduction, reinforcement learning, etc. The key problem in analyzing document collections at multiple levels is to find a multiscale embedding of the documents. This problem can be formalized as follows: given a collection of documents, each of which is represented as a bag of words, can we discover a hierarchical representation of the documents that reveals multiscale conceptual regularities?

Topic models are an important tool to find concepts from document corpora. They have been successfully used to analyze large amounts of textual information for many tasks. A topic could be thought as a multinomial word distribution learned from a collection of documents using either linear algebra or statistical techniques. The words that contribute more to each topic provide keywords that briefly summarize the themes in the collection. The new representations of documents can be computed by projecting the original documents onto the space (topic space) spanned by topic vectors. Popularly used topic models include the aforementioned LSI and Latent Dirichlet Allocation (LDA) [Blei, Ng, & Jordan, 2003]. However, these models can only find regularities at a single level. Recently, several statistical approaches were proposed to find topical hierarchies. One of them is hLDA [Blei *et al.*, 2004]. Such new methods heavily depend on detailed prior information, like number of levels, number of topics. Exact inference in these graphical models is also generally intractable, and requires sampling-based methods.

In this paper, we present a new diffusion model based approach that automatically and efficiently finds multiscale embeddings of documents in a given corpus. Our method builds on recent work in harmonic analysis, in particular *diffusion wavelets* [Coifman & Maggioni, 2006]. Harmonic analysis is a well-studied area of mathematics, which includes Fourier analysis in continuous spaces. Recent work in harmonic analysis has turned to wavelet methods, which produce a multiscale analysis of functions at many temporal and spatial levels. Diffusion wavelets (DWT) is a recent extension of wavelet methods to functions on discrete spaces like graphs. Unlike classical wavelets, in diffusion wavelets the basis functions at each level of the hierarchy are not predetermined, but need to be constructed by dilation using the dyadic powers of a diffusion matrix. One novel aspect of our work is that the DWT multiscale construction is on the variables, but not on the instances (as many previous applications of DWT have been). The key strength of the approach is that it is completely data-driven, largely parameter-free and can automatically determine the number of levels of the topical hierarchy, as well as the topics at each level. To our knowledge, none of the competing methods can produce a multiscale analysis of this type. Further, when the input term-term matrix is a "local" diffusion operator, the algorithm runs in time approximately linear in the number of non-zero elements of the matrix. In contrast to the topics learned from another linear algebra based method LSI, our topics have local support. This is particularly useful when the concept only involves a small group of words. We achieve multiscale embeddings of document corpora by projecting the documents onto such a hierarchical, interpretable topic space.

Our approach is tested on three real world data sets: the NIPS (1-12) conference full paper data set, which is available at www.cs.toronto.edu/∼roweis/data.html, the 20 NewsGroups data set (people.csail.mit.edu/jrennie/20Newsgroups) and the TDT2 data set (projects.ldc.upenn.edu/TDT2). The results show that the multiscale diffusion model can successfully identify the structure of each collection at multiple scales.

2 Learning Topic Spaces

Learning a topic space means learning the topic vectors spanning the concept space. In a collection of documents (defined on a vocabulary with n terms), any document can be represented as a vector in R^n, where each axis represents a term. The ith element of the vector can be some function of the number of times that the ith term occurs in the document. There are several possible ways to define the function to be used here (frequency, tf-idf, etc.), but the precise method does not affect our results. In this paper, we assume A is an $n \times m$ matrix whose rows represent terms and columns represent documents.

2.1 Learning Topic Spaces using LDA

Latent Dirichlet Allocation (LDA) [Blei, Ng, & Jordan, 2003] is a widely used probabilistic topic model and the basis for many variants. LDA treats each document as a mixture of topics, where each topic is a distribution over words in a vocabulary. To generate a document, LDA first samples a per-document distribution over topics from a Dirichlet distribution, and then it samples a topic from the distribution and a word from the topic. Documents in LDA are linked only through a single Dirichlet prior, so the model makes no attempt to find the distribution over topic mixtures. LDA is a "flat" topic model.

2.2 Learning Topic Spaces using hLDA and others

The hLDA model [Blei *et al.*, 2004] represents the distribution of topics within documents by organizing the topics into a tree. Each document is generated by the topics along a path of this tree. To learn the model from the data, we need to alternately sample between choosing a new path through the tree for each document and assigning each word in each document a topic along the chosen path. In the hLDA model, the quality of the distribution of topic mixtures depends on the topic tree. To learn the structure of the tree, hLDA applies a nested Chinese restaurant process (NCRP), which requires two parameters: the number of levels of the tree and a parameter γ. hLDA and some other methods can learn hierarchical topics, but they need detailed prior information, such as number of levels, number of topics and the performance of these models heavily depends on the priors. Inference in these graphical models is also generally intractable, and typically a sampling based approach is used to train these models, which is computationally expensive.

2.3 Learning Topic Spaces using LSI

Latent semantic indexing (LSI) [Deerwester *et al.*, 1990] is a well-known linear algebraic method to find topics in a text corpus. The key idea is to map high-dimensional vectors to a lower dimensional representation in a latent semantic space. The goal of LSI is to find a mapping that provides information that reveals semantical relations between the entities of the interest. Let the singular values of A be $\delta_1 \geq \cdots \geq \delta_r$, where r is the rank of A. The singular value decomposition of A is $A = U\Sigma V^T$, where $\Sigma = diag(\delta_1, \cdots \delta_r)$, U is an $n \times r$ matrix whose columns are orthonormal, and V is an $m \times r$ matrix whose columns are also orthonormal. LSI constructs a rank-k approximation of the matrix by keeping the k largest singular values in the above decomposition, where k is usually much smaller than r. LSI is also a "flat" topic model, which means it cannot find hierarchical topics.

The term-term matrix AA^T gives the correlation between terms over the documents. $AA^T = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma\Sigma^T U^T$, so the column vectors of U (topic vectors) are also the eigenvectors of the term-term matrix AA^T.

2.4 Learning Topic Spaces using Diffusion Models

The term-term matrix AA^T is a *Gram* matrix with nonnegative entries. Define D as a diagonal matrix, whose entry D_{ii} is the sum of the entries on the i-th row of AA^T. We define the normalized term-term matrix T as $D^{-0.5}AA^TD^{-0.5}$. In fact, the *normalized Laplacian operator* associated with AA^T is $\mathcal{L} = I - T$. The Laplacian matrix has become a cornerstone of recent methods in machine learning, in areas ranging from clustering, semi-supervised learning and dimensionality reduction. Instead of learning the eigenvectors of T, multiscale diffusion analysis involves learning the diffusion *scaling functions* of T using diffusion wavelets [Coifman & Maggioni, 2006]. This process can be interpreted geometrically as projecting data to lower dimensional spaces by using the scaling functions while preserving the large scale information which is inherent in the data. The method provides a multiscale embedding, which means it automatically reveals the geometric structure of the data at different scales. The subspace spanned by diffusion scaling functions from T is exactly the subspace spanned by certain eigenvectors of T (with largest eigenvalues) up to a precision ε [Coifman & Maggioni, 2006]. However, the diffusion scaling functions are multiscale basis functions, with local support and can be computed efficiently. These properties make our multiscale diffusion approach attractive in applications to text mining. A detailed description is in Sec 4.

3 Finding a Multiscale Embedding of the Documents from a Corpus

If a topic space $\mathcal{S}$ is spanned by a set of r topic vectors, we write the set as $S = (t(1), \cdots, t(r))$, where topic $t(i)$ is a column vector $(t(i)_1, t(i)_2 \cdots, t(i)_n)^T$. Here n is the size of the vocabulary set, $\|t(i)\| = 1$ and the value of $t(i)_j$ represents the contribution of term j to $t(i)$. Obviously, S is an $n \times r$ matrix. We know the term-document matrix A (an $n \times m$ matrix) models the corpus, where m is the number of the documents and columns of A represent documents in the "term" space. The low dimensional embedding of A in the "topic" space $\mathcal{S}$ is then $A_{Topic} = S^T A$. A_{Topic} is an $r \times m$ matrix, whose columns are the new representations of documents in $\mathcal{S}$.

A diffusion model extracts topics at multiple scales, yielding a multiscale embedding of the documents. The new representation of the documents at a particular scale may significantly compress the data preserving the most useful information at that scale. Since all the topics are interpretable, we may read the topics at different scales and select the best scale for embedding. At one scale, we can determine which topic is more relevant to our task and discard the non-useful topics. Our diffusion model based multiscale embedding method provides a very powerful tool to analyze the document corpora and will be quite useful for classification, information retrieval, clustering, etc. Later, we will show that it is also efficient.

4 The Main Algorithm

In the main algorithm below and Figure 1, the notation $[T]_{\phi_a}^{\phi_b}$ denotes the matrix representation of operator T, where the column space or range of T is represented with respect to the basis ϕ_b, and the row space or input space is represented using the basis ϕ_a. The subscripts in ϕ_b or ϕ_a denote the scale. The notation $[\phi_b]_{\phi_a}$ denotes basis ϕ_b written on the basis ϕ_a.

4.1 The Algorithmic Procedure

Assume the term-document matrix A is already given. The algorithmic procedure is stated below (some of the notation given below is explained in Section 4.3):

1. **Constructing the normalized term-term matrix T:** $T = D^{-0.5} A A^T D^{-0.5}$, where D is a diagonal matrix, whose entry D_{ii} is the sum of the entries on the i-th row of AA^T.

2. **Generating Diffusion Models:** $\{\phi_j, \psi_j\} = DWT(T, I, QR, J, \varepsilon)$.
   - I is an identify matrix; J is the max step number; ε is the desired precision. QR is a modified QR decomposition [Coifman & Maggioni, 2006].
   - ϕ_j: diffusion scaling functions at level j. ψ_j: wavelet functions at level j.

3. **Computing the extended basis functions:**
   - $[\phi_j]_{\phi_0}$, the representation of the basis functions at level j in the original space, is computed as follows: $[\phi_j]_{\phi_0} = [\phi_j]_{\phi_{j-1}} [\phi_{j-1}]_{\phi_{j-2}} \cdots [\phi_1]_{\phi_0} [\phi_0]_{\phi_0}$.
   - $[\phi_j]_{\phi_0}$ is an $n \times n_j$ matrix. Each column vector represents a topic at level j. Entry k on the column vector shows term k's contribution to this topic.

4. **Computing multiscale embeddings of the corpora:** At scale j, the embedding of A is $([\phi_j]_{\phi_0})^T A$.

```
{φ_j, ψ_j} = DWT(T, φ_0, QR, J, ε)
 //φ_j: "Scaling" basis functions at scale j.
 //ψ_j: "Wavelet" basis functions at scale j.
 //QR: A function computing a sparse QR decomposition.
 //J: Max number of steps to compute. ε: Precision.
 //*: Transpose conjugate.
For j = 0 to J − 1 {
    ([φ_{j+1}]_{φ_j}, [T^{2^j}]_{φ_j}^{φ_{j+1}}) ← QR([T^{2^j}]_{φ_j}^{φ_j}, ε);
    [T^{2^{j+1}}]_{φ_{j+1}}^{φ_{j+1}} = ([T^{2^j}]_{φ_j}^{φ_{j+1}})([T^{2^j}]_{φ_j}^{φ_{j+1}})*;
    [ψ_j]_{φ_j} ← QR(I_{<φ_j>} − [φ_{j+1}]_{φ_j}[φ_{j+1}]*_{φ_j}, ε);
}
```

Figure 1: The DWT Procedure. J can be omitted, since the representation of T will converge to a scalar value at some level, and the construction will terminate.

4.2 High Level Explanation

Instead of using the document-document matrix $A^T A$, as is done in [Coifman & Maggioni, 2006], we run the multiscale algorithm on the term-term matrix AA^T, which models the co-occurrence relationship between any two term vectors over the documents in the given corpora. In fact, almost all state of the art approaches learn topics from such co-occurrence information. Our algorithm starts with the normalized term-term co-occurrence matrix and then repeatedly applies QR decomposition to learn the topics at the current level while at the same time modifying the matrix to focus more on low-frequency indirect co-occurrences for the next level. Our approach is in spirit similar to LSI, but goes beyond LSI to naturally generate topics in multiple resolutions through progressively constructing a matrix to model the low frequency indirect co-occurrences.

4.3 The DWT Procedure

The diffusion wavelets algorithm is summarized in Figure 1. Assume at an arbitrary scale i, we have n_i basis functions, and length of each function is l_i, then $[T]_{\phi_a}^{\phi_b}$ is an $n_b \times l_a$ matrix, $[\phi_b]_{\phi_a}$ is an $l_a \times n_b$ matrix. The scaling function $[\phi_j]_{\phi_{j-1}}$ plays a major role in this paper since it provides a mapping between the data at coarse scale and fine scale spaces. We can represent basis functions at level j in terms of the basis functions at the next lower level. In this manner, the extended basis functions can be expressed in terms of the original bases as $[\phi_j]_{\phi_0} = [\phi_j]_{\phi_{j-1}} [\phi_{j-1}]_{\phi_0}$, so we can compute $[\phi_j]_{\phi_0}$ using $[\phi_j]_{\phi_0} = [\phi_j]_{\phi_{j-1}} [\phi_{j-1}]_{\phi_{j-2}} \cdots [\phi_1]_{\phi_0} [\phi_0]_{\phi_0}$. Each element on the right hand side of the equation is created in the DWT procedure. The elements in $[\phi_j]_{\phi_0}$ are usually much coarser and smoother than the initial elements in ϕ_0, which is why they can be represented in compressed form. Given $[\phi_j]_{\phi_0}$, any function on the compressed large scale space can be extended naturally to the original space or vice versa. The connection between any vector in the original space and its compressed representation at scale j is $v_{[\phi_j]} = ([\phi_j]_{\phi_0})^T v_{[\phi_0]}$.

The DWT procedure is as follows: the original matrix T represents the one step transition probability between data points ("terms" for our case). The QR subroutine is a Gram-Schmidt orthogonalization routine that finds the QR decomposition up to precision ε (at scale j), while filtering out the "high frequency" noise. Then we construct the basis functions from the new matrix. We usually have a smaller number of basis functions to characterize the new matrix, since a lot of high frequency information has already been filtered out. We use the low frequency information to compute the two time step transition from T^{2^j} resulting in a new representation of $T^{2^{j+1}}$ at the next level. The matrix of T can be thought as a transition matrix, and the probability of transition from x to y in j time steps is given by $T^j(x, y)$. So the procedure described in Figure 1 is equivalent to running the Markov chain forward in time and allows us to integrate the local geometry and therefore reveal the relevant geometric structures of data at different scales. At scale j, the representation of T^{2^j} is compressed based on the amount of remaining information and the precision we want to keep.

4.4 Comparison to Other Methods

As shown in Figure 1, the spaces at different levels are spanned by a different number of basis functions. These numbers reveal the dimensions of the relevant geometric structures of data at different levels. These numbers are completely data-driven: the diffusion-wavelet approach can automatically find the number of levels and simultaneously generate the number of topics at each level. In fact, once the term-document matrix A is given, users only need to specify one parameter ε – the precision. In fact, this parameter can be automatically set by computing the average of the non-zero entries on the normalized term-term matrix T, and taking its product with a small number like 10^{-5} to get ε. So, our approach is essentially parameter free, a significant advantage over competing methods. To incorporate prior knowledge, the term-document matrix A can be suitably modified.

Learning hierarchical topics could be done in almost linear time, when T is a "local" diffusion operator [Coifman & Maggioni, 2006; Maggioni & Mahadevan, 2006]. The main idea is that most examples defined in the diffusion operator have "small" degrees in which transitions are allowed only among neighboring points, and the spectrum of the transition matrix decays rapidly. This result is in contrast to the time needed to compute k eigenvectors, which is $O(kn^2)$. In many applications, the normalized term-term matrix T is already a localized diffusion operator. If it is not, we can simply convert it to such a matrix: for each term in the collection, we only consider its most relevant k terms since the relationships between terms that co-occur many times are more important. The same technique has been widely used in manifold learning to generate the relationship graph from the given data examples. The algorithm is modified to retain the top k entries in each row of T, and all other entries are set to 0. The resulting matrix is not symmetric, so we need to symmetrize it in the end.

The space spanned by topic vectors from diffusion models are the same as the space spanned by some LSI topic vectors up to a precision ε. However, the topic vectors (in fact eigenvectors) from LSI have a potential drawback that they detect only global smoothness, and may poorly model the concept/topic which is not globally smooth but only piecewise smooth, or with different smoothness in different regions. Unlike the global nature of eigenvectors, our topic vectors are local (sparse). This can better capture some concepts/topics that only involve a particular group of words. Experiments show that most diffusion model based topics are interpretable, such that we can interpret the topics at different scales and select the best scale for embedding. Further, at the selected scale, we can check which topic is more relevant to our application and skip the non-useful topics. In contrast, many LSI topics are not interpretable.

Topic vectors from diffusion models are orthonormal to each other. In other words, for any two topics t_i and t_j at an arbitrary level, we have $t_i \cdot t_j = 0$ and $\|t_i\| = \|t_j\| = 1$. This means the information encoded using diffusion model topics is not redundant and representation of documents in the topic space is unique. This property does not hold for parametric statistical approaches (like LDA, hLDA).

The complexity of generating a diffusion model mostly depends on the size of the vocabulary set in the corpus, but not the number of the documents, or the number of the tokens. We know no matter how large the corpus is, the size of the vocabulary set is determined. So our approach should be scalable to large data sets.

5 Experimental Results

In this section, we describe the results of multiscale analysis on three real world data sets. We use the NIPS conference paper data set to show what the resulting multiscale analysis look like, and how to interpret these topics. We use the 20 NewsGroups data and TDT2 data to show the multiscale embeddings of the corpora.

Since our model is parameter-free, we do not need any special settings. The precision we used for all these experiments was 10^{-5}. One problem that is important but we have not addressed so far is how to interpret topics learned from our diffusion models. For any given topic vector v, we know it is a column vector of length n, where n is the size of the vocabulary set and $\|v\| = 1$. The entry $v[i]$ represents the contribution of term i to this topic. To explain the main concept of topic v, we sort the entries on v and print out the terms corresponding to the top 10 entries. These terms summarize the topics in the collection.

5.1 NIPS Paper

We generated hierarchical topics from the NIPS paper data set, which includes 1,740 papers. The original vocabulary set has 13,649 terms. The corpus has 2,301,375 tokens in total. We filtered out the terms that appear ≤ 100 times in the corpus, and only 3,413 terms were kept. The collection did not change too much. The number of the remaining tokens was 2,003,017. For comparison purpose, we also tested LSI, LDA and hLDA using the same data set.

Table 1: Number of topics at different levels (diffusion model, NIPS)

Level	Number of Topics
1	3413
2	1739
3	1052
4	37
5	2

Table 2: Some topics at level 4 (diffusion model, NIPS)

Top 10 Terms
policy state action reinforcement actions learning reward mdp agent sutton
mouse chain proteins region heavy receptor protein alpha human domains
distribution data gaussian density bayesian kernel posterior likelihood em regression
chip circuit analog voltage vlsi transistor charge circuits gate cmos
speech hmm word speaker phonetic recognition spike markov mixture acoustic
iiii border iii texture ill bars suppression ground bar contextual
face facial images faces image tangent spike object views similarity
adaboost margin boosting classifiers head classifier hypothesis training svm motion
stress syllable song heavy linguistic vowel languages primary harmony language
routing traffic load shortest paths route path node message recovery
actor critic pendulum tsitsiklis pole barto harmony signature routing instructions
documents query document retrieval queries words relevant collection text ranking
classifier classifiers clause knn rbf tree nearest neighbor centers classification
stack symbol strings grammars string grammar automata grammatical automaton giles
song template production kohonen syllable pathway harmonic nucleus lesions motor

Multiscale Diffusion Model identifies 5 levels of topics, and the number of the topics at each level is shown in Table 1. At the first level, each column in T is treated as a topic. At the second level, the number of the columns is almost the same as the rank of T. At level 4, number of topics goes down to a reasonable number 37. Finally at level 5, the number of topics is 2. The 2 topics at level 5 are "*network, learning, model, neural, input, data, time, function, figure, set*" and "*cells, cell, neurons, firing, cortex, synaptic, visual, stimulus, cortical, neuron*". Obviously, the first is about machine learning, while the second is about neuroscience. These two topics are exactly the real topics at the highest level of NIPS. Almost all 37 topics at level 4 look semantically meaningful. They nicely capture the function words. Some examples are in Table 2.

LSI computes "flat" topics only, so we compare the top 37 LSI topics to the results from diffusion model. The LSI topics (not shown here) look much worse. The reason is the diffusion model based topics are with local support, while LSI topics are globally "smooth". Even though such vectors are spanning the same space, they look quite different. "Local support" is particularly important to represent a concept that only involve a small number of words in document corpora.

LDA was also tested on this data set. To use LDA, we need to specify the number of topics. In this test, we tried two numbers: 2 and 37. When topic number is 2, the two topics are "*model, network, input, figure, time, system, neural, neurons, output, image*" and "*learning, data, training, network, set, function, networks, algorithm, neural, error*". They do not cover neuroscience, which is covered by our diffusion model. Given the space constraint, we did not list the 37 LDA topics (most of them also look reasonable). Again, to use LDA, users need to specify the number of topics, but in diffusion model, we automatically learn this number.

hLDA requires specifying the number of levels of the topic tree (and some other parameters). In this test, we set this number to 3. The *hLDA* module in MALLET (mallet.cs.umass.edu) was applied for this task. The resulting topic tree is in Table 3, where the $Node$ record shows the path from the root to the node. Node 1 is the root of the tree, which has 3 children (1.1, 1.2 and 1.3). Both Node 1.1 and 1.2 have two children. Node 1.3 has one child: 1.3.1. hLDA does not cover the topic of neuroscience at level 1 and 2. Compared to diffusion models, hLDA topics are harder to interpret. We also tested level=4, and the result did not make much difference.

Table 3: hLDA topics (NIPS)

Node	Top 10 Terms
1	task performance training data learning output algorithm time processing trained
1.1	function terms networks abstract linear case references equation set functions
1.1.1	activity brain visual response neurons cells shown model cell properties
1.1.2	posed cell soc movements contour response minimization biol orientations perpendicular
1.2	statistical distribution figure matrix approach parameters gaussian data model methods
1.2.1	neuron cells neurons synaptic inhibition cell physics american phase strength
1.2.2	function theory show result finite class called positive introduction define
1.3	finite noise gaussian constant solved terms corresponds equation exp variables
1.3.1	og obtain equations dynamics distribution matrix choice stable moore estimation

Empirical Evaluation of Time Complexity

Given the collection with 2,003,017 tokens, the multiscale diffusion model needs roughly 15 minutes (2G PC with 2G memory) to do the multiscale analysis. This includes data preparation, construction of the diffusion model and computing topic vectors at all 5 levels. In contrast, we need about 4 and 6 minutes to compute 37 topics using LSI and LDA on the same machine. LSI and LDA only computes "flat" topics, but not topics at multiple levels, and they do not need to explore the intrinsic structure of the data set, so they are doing something much simpler. Running hLDA is much more expensive than the others. It needs roughly 20 hours for this task. Considering the time complexity, hLDA is not tested in Sec 5.2 and 5.3.

5.2 20 NewsGroups

The 20 NewsGroups data set is a popular data set for experiments in text applications. The version that we are using is a collection of 18,774 documents (11,269 for training, 7,505 for testing), partitioned evenly across 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other, while others are highly unrelated. The data set has 61,188 terms in the vocabulary set (stop words are not removed) and nearly 2,500,000 tokens. We filtered out the terms that appear ≤ 100 times in the training set, and only 2,993 terms were kept.

Using the training data, diffusion model identifies 5 levels of topics, and the number of topics at each level is: 2993, 2992, 589, 29 and 1. Since 29 is the closest number to the real topic number 20, we pick up level 4 for further analysis. We find 3 of the 29 topics are related to stop words. For example, the top 10 words of one such topic are: "*the, to, of, and, in, is, that, it, for, you*". The remaining 26 topics cover almost all 20 known topics. For example, the topic "*probe, mars, lunar, moon, missions, surface, jupiter, planetary, orbit, planet*" corresponds to topic "*space*". LDA and LSI were also tested. For LDA, we tried two topic numbers: 20 and 29. The number of 29 returned a better result. The LDA topics do not look as good as the topics from the diffusion

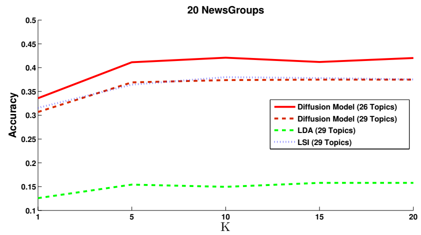

Figure 2: Classification results with different embeddings.

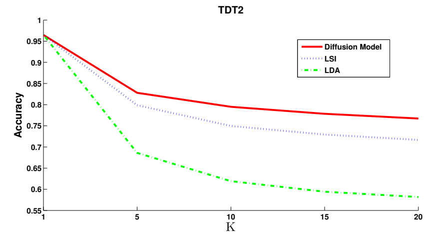

Figure 3: Classification results with different embeddings

model. Stop words always dominate the top words of each topic. For example, the topic "*the, and, of, to, for, key, space, on, in, by*" might be related to topic "*space*", but most of the top words are stop words. The LSI topics do not look good either. For many applications, LSI topics might span a good concept space, but they are hard to interpret.

To compare the low dimensional embeddings generated from the diffusion model with LSI and LDA, we used a kNN method to classify the test documents. We first represent all the documents in the topic space using the 29 topics learned from the training set. For each test document, we compute the similarity (dot product) of it and all the training documents. For each news group, we consider the top k most similar documents to the test document. The label of the group with the largest sum of such similarities is used to label the test document. Since 3 topics returned by our diffusion model are related to stop words, we also ran a test using the remaining 26 topics. We tried different k in the experiment and the results are shown in Figure 2. From the figure, it is clear that the embeddings coming from diffusion model (29 topics) and LSI are similar. Both of them are better than the embedding from LDA. It is also shown that filtering out the non-relevant topics can improve the performance. The LSI topics are hard to interpret, so we can not filter any of them out.

5.3 TDT2

The TDT2 corpus consists of data collected during the first half of 1998 and taken from 6 sources, including 2 newswires (APW, NYT), 2 radio programs (VOA, PRI) and 2 television programs (CNN, ABC). It consists of more than 10,000 documents which are classified into 96 semantic categories. In the data set we are using, the documents that appearing in more than one category were removed, and only the largest 30 categories were kept, thus leaving us with 9,394 documents in total. Using the same procedure shown in the other tests, we identified a 5 level hierarchy (topic number at each level is: 2800, 2793, 287, 17, 2). To better understand what the embeddings look like, we project the documents onto a 3D space spanned by three topic vectors from each model (Diffusion model: top 3 topic vectors at level 4; LDA: all topics when topic number =3; LSI: top 3 topic vectors). In this test, we plot the documents from category 1-7 (nearly 7,000 documents in total) and each color represents one category. The diffusion model returns the best embedding (Figure 4). We also run a leave one out test with kNN method (as described in the 20 NewsGroups test) to classify each document in the collection. The results are in Figure 3. It is also clear that the embedding from the multiscale diffusion model is better than LSI and LDA.

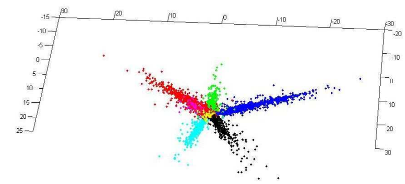

Figure 4: 3D embedding of TDT2 (diffusion model)

6 Conclusions

In this paper, we propose a multiscale diffusion model to extract semantic structure of real-world corpora of text documents at multiple scales. Experimental results show this approach successfully extracts hierarchical regularities at multiple levels. The hierarchy yields semantically meaningful topics, and efficient multiscale embeddings for classification. The same technique can also be applied to social network analysis and author-topic modeling.

Acknowledgments

Support for this research was provided in part by the National Science Foundation under grants IIS-0534999 and IIS-0803288. We thank Xuerui Wang for helpful discussions, David Mimno and Limin Yao for help on hLDA.

References

[Blei *et al.*, 2004] Blei, D.; Griffiths, T.; Jordan, M.; and Tenenbaum, J. 2004. Hierarchical topic models and the nested Chinese restaurant process. In *Advances in Neural Information Processing Systems*.

[Blei, Ng, & Jordan, 2003] Blei, D.; Ng, A.; and Jordan, M. 2003. Latent Dirichlet allocation. In *Journal of Machine Learning Research*, volume 3, 993–1022.

[Coifman & Maggioni, 2006] Coifman, R., and Maggioni, M. 2006. Diffusion wavelets. *Applied and Computational Harmonic Analysis* 21:53–94.

[Deerwester *et al.*, 1990] Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41(6):391–407.

[Maggioni & Mahadevan, 2006] Maggioni, M., and Mahadevan, S. 2006. Fast direct policy evaluation using multiscale analysis of Markov diffusion processes. *ICML*.