Chapter 19

Topic Modeling for Large-Scale Multimedia Analysis and Retrieval

Juan Hu, Yi Fang, Nam Ling, and Li Song

Contents
19.1 Introduction 376
19.2 Large-Scale Computing Frameworks 377
19.3 Probabilistic Topic Modeling 379
19.4 Couplings among Topic Models, Cloud Computing, and Multimedia Analysis 382
  19.4.1 Large-Scale Topic Modeling 382
  19.4.2 Topic Modeling for Multimedia 384
  19.4.3 Large-Scale Computing in Multimedia 385
19.5 Large-Scale Topic Modeling for Multimedia Retrieval and Analysis 386
19.6 Conclusions and Future Directions 388
References 389

Abstract
The explosion of multimedia data in social media raises a great demand for developing effective and efficient computational tools to facilitate producing, analyzing, and retrieving large-scale multimedia content. Probabilistic topic models prove to be an effective way to organize large volumes of text documents, while much fewer related models are proposed for other types of unstructured data such as multimedia content, partly due to the high computational cost. With the emergence of cloud computing, topic models are expected to become increasingly applicable to multimedia data. Furthermore, the growing demand for a deep understanding of multimedia data on the web drives the development of sophisticated machine learning methods. Thus, it is greatly desirable to develop topic modeling approaches to multimedia applications that are consistently effective, highly efficient, and easily scalable. In this chapter, we present a review of topic models for large-scale multimedia analysis. Our goal is to show the current challenges from various perspectives and to present a
comprehensive overview of related work that addresses these challenges. We will also discuss several research directions in the field.

19.1 INTRODUCTION

With the arrival of the big data era, recent years have witnessed an exponential growth of multimedia data, thanks to the rapid increase of processor speed, cheaper data storage, prevalence of digital content capture devices, as well as the flooding of social media like Facebook and YouTube. New data generated each day have reached 2.5 quintillion bytes as of 2012 (Dean and Ghemawat 2008). Particularly, more than 10 h of videos are uploaded onto YouTube every minute, and millions of photos are available online every week. The explosion of multimedia data in social media raises great demand in developing effective and efficient computational tools to facilitate producing, analyzing, and retrieving large-scale multimedia content. Big data analysis for basic tasks such as classification, retrieval, and prediction has become ever popular for multimedia sources in the form of text, graphics, images, audio, and video. The data set is so large and noisy that the scalability of the traditional data mining algorithms needs to be improved. The MapReduce framework designed by Google is very simple to implement and very flexible in that it can be extended for various large-scale data processing functions. This framework is a powerful tool to develop scalable parallel applications to process big data on large clusters of commodity machines. The equivalent open-source Hadoop MapReduce developed by Yahoo is now very popular in both the academic community and industry.

In the past decade, much effort has been made in the information retrieval (IR) field to find lower-dimensional representation of the original high-dimensional data, which enables efficient processing of a massive data set while preserving essential features. Probabilistic topic models proved to be an effective way to organize large volumes of text documents. In natural language processing, a topic model refers to a type of statistical model for representing a collection of documents by discovering abstract topics. At the early stage, a generative probabilistic model for text corpora is developed to address the issues of TF-IDF, namely, that the dimension reduction effect using TF-IDF is rather small (Papadimitriou et al. 1998). Later, another important topic model named probabilistic latent semantic indexing (PLSI) was created by Thomas Hofmann in 1999. Essentially, PLSI is a two-level hierarchical Bayesian model where each word is generated from a single topic and each document is reduced to a probability distribution of a fixed set of topics. Latent Dirichlet allocation (LDA) (Blei et al. 2002) is a generalization of PLSI developed by providing a probabilistic model at the level of documents, which avoids the serious over-fitting problem as the number of parameters in the model does not grow linearly with the size of the corpus. LDA is now the most common topic model, and many topic models are generally an extension of LDA by relaxing some of the statistical assumptions. The probabilistic topic model exemplified by LDA aims to discover the hidden themes running through the words that can help us organize and understand the vast information conveyed by massive data sets.

To improve the scalability of a topic model for big data analysis, much effort has been put into large-scale topic modeling. Parallel LDA (PLDA) was designed by distributing...
Gibbs sampling for LDA on multiple machines (Chang et al. 2009). Another flexible large-scale topic modeling package named Mr.LDA is implemented in MapReduce, where model parameters are estimated by variational inference (Zhai et al. 2012). A novel architecture for a parallel topic model is demonstrated to yield better performance (Smola and Narayanamurthy 2010).

While topic models are proving to be effective methods for corpus analysis, much fewer related models have been proposed for other types of unstructured data such as multimedia content, partly due to high computational cost. With the emergence of cloud computing, topic models are expected to become increasingly applicable to multimedia data. Furthermore, the growing demand for a deep understanding of multimedia data on the web drives the development of sophisticated machine learning methods. Thus, it is greatly desirable to develop topic modeling approaches for large-scale multimedia data that are consistently effective, highly efficient, and easily scalable. In this chapter, we present a review of topic models for large-scale multimedia analysis.

The chapter is organized as follows. Section 19.2 gives an overview of several distributed large-scale computing frameworks followed by a detailed introduction of MapReduce framework. In Section 19.3, topic models are introduced, and we present LDA as a typical example of a topic model as well as inference techniques such as Gibbs sampling and variational inference. More advanced topic models are also discussed in this section. A review of recent work on large-scale topic modeling, topic modeling for multimedia analysis, and large-scale multimedia analysis is presented in Section 19.4. Section 19.5 demonstrates recent efforts done on large-scale topic modeling for multimedia retrieval and analysis. Finally, Section 19.6 concludes this chapter.

19.2 LARGE-SCALE COMPUTING FRAMEWORKS

The prevalence of large-scale computing drives parallel processing with thousands of computer nodes compared to most computing in the past done on a single processor. Academic research areas and industries have been strongly connected in this area as lots of companies such as Google, Microsoft, Yahoo, Twitter, and Facebook face the technical challenge of efficiently and reliably dealing with ever-growing data sets. These companies, in turn, created many innovative large-scale computing frameworks. For example, Google’s MapReduce (Dean and Ghemawat 2004) is perhaps the most popular distributed framework, and the open-source version of MapReduce created by Yahoo with Apache Hadoop can be seen anywhere nowadays. There are also many other successful frameworks such as Microsoft’s Dryad (Isard et al. 2007), Yahoo’s S4 for unbounded streaming data, Storm used by Twitter, and so on. As most of these frameworks can be considered as extensions of MapReduce with enhancements for specific applications, in the rest of the section, the large-scale computing framework is exemplified and illustrated by MapReduce.

MapReduce is a framework for processing massive data sets in a parallel diagram across a large number of computer nodes. The computer nodes can be a cluster where they are on the same local network and use similar hardware, or a grid where computer nodes are shared across geographically and administratively distributed systems and use more heterogeneous hardware. In the parallel computing architecture, computer nodes are stored
on racks, roughly 8 to 64 for each rack. The computer nodes on the same rack are connected by gigabyte Ethernet, while the racks are interconnected by a switch.

There are two kinds of computer nodes in this framework, the master node and the worker node. The master node basically assigns tasks for worker nodes and keeps track of the status of the worker nodes, which can be idle, executing a particular task, or finished completed it. Thus the master node takes central control role of the whole process and failure at master node can be disastrous. The entire process could be down, and all the tasks need to be restarted. On the other hand, failure in the worker node can be detected by the master node as it periodically pings the worker processes. The master node can manage the failures at the worker nodes, and all the computing tasks will be complete eventually.

The name of MapReduce comes naturally from the essential two functions under this framework, that is, the Map step and the Reduce step, as described in Figure 19.1. The input of the computing process is chunks of data, which can be any type, such as documents or tuples. The Map function converts input data into key-value pairs. Then the master controller chooses a hash function that is applied to keys in the Map step and produces a bucket number, which is the total number of Reduce tasks defined by a user. Each key in the Map task is hashed, and its key-value pair is put in the local buckets by grouping and aggregation, each of which is destined for one of the Reduce tasks. The Reduce function takes pairs consisting of a key and its list of associated values combined in a user-defined way. The output of a Reduce task is a sequence of key-value pairs, which consists of the key received from the Map task and the combined value constructed from the list of values that the Reduce task received along with the key. Finally, the outputs from all the Reduce tasks are merged into a single file.

The MapReduce computation framework can be best illustrated with a classic example of word count. Word count is very important as it is exactly the term frequency in the IR model. The input file for the framework is a corpus of many documents. The words are the keys of the Map function, and the count of occurrences of each word is the value corresponding to the key. If a word $w$ appears $m$ times among all the documents assigned to that process, in the Reduce task, we simply add up all values so that after grouping and aggregation, the output is a sequence of pairs $(w, m)$.

In the real MapReduce execution, as shown in Figure 19.1, a worker node can handle either a Map task or a Reduce task but will be assigned only one task at a time. It is

![FIGURE 19.1  Schematic of MapReduce computation.](image-url)
reasonable to have a smaller number of Reduce tasks compared to Map tasks as it is necessary for each Map task to create an intermediate file for each Reduce task and if there are too many Reduce tasks, the number of intermediate files explodes. The original implementation of the MapReduce framework was done by Google via Google File System. The open-source implementation by Yahoo was called Hadoop, which can be downloaded along with the Hadoop file system, from the Apache foundation. Computational processing can occur on data stored either in a distributed file system or in a database. MapReduce can take advantage of locality of data, processing data on or near the storage assets to decrease transmission of data with a tolerance of hardware failure.

MapReduce allows for distributed processing of the map and reduction operations. Provided each mapping operation is independent of the others, all maps can be performed in parallel, though in practice, it is limited by the number of independent data sources and the number of CPUs near each source. Similarly, a set of “reducers” can perform the reduction tasks. All outputs of the map operation that share the same key are presented to the same reducer at the same time. While this process can often appear inefficient compared to algorithms that are more sequential, MapReduce can be applied to significantly larger data sets than “commodity” servers can handle. A large server farm can use MapReduce to sort petabytes of data in only a few hours. The parallelism also offers some possibility of recovering from partial failure of servers or storage during the operation: If one mapper or reducer fails, the work can be rescheduled as long as the input data are still available.

While the MapReduce framework is very simple, flexible, and powerful, the data-flow model is ideally suited for batch processing of on-disk data, where the latency can be very poor and its scalability to real-time computation is limited. To solve these issues, Facebook’s Puma and Yahoo’s S4 are proposed for real-time aggregation of unbounded streaming data. A novel columnar storage representation for nested records was proposed in the Dremel framework (Melnik et al. 2010), which improves the efficiency of MapReduce. In-memory computation was allowed in another data-centric programming model. Piccolo (Powell and Li 2010) and Spark, which were created by the Berkeley AMPLab, have been demonstrated to be able to deliver much faster performance. Pregel was proposed for large-scale graphical processing (Malewicz et al. 2010).

19.3 PROBABILISTIC TOPIC MODELING

The motivation of topic models comes from tuition in real life. Suppose you want to find books about tennis at a public library. You would directly go to the bookshelves labeled “sports.” You can do this because the librarian has arranged books into categories by the topics inside the books. However, with the exponentially growing information available online, it is impossible to hire enough human power to read and label web pages, images, or videos with their topics; it is highly desirable to label the large-scale information automatically so that we can easily find what we are looking for. Probabilistic topic models provide powerful tools to discover the hidden themes running through large-scale texts, images, and videos without any prior annotations or label information.

Latent semantic indexing (LSI) utilizes the singular value decomposition of the document term frequency (Deerwester et al. 1990), which provides a solution for high-dimensional
big data sets by giving shorter and lower-dimensional representation of original documents. pLSI took a remarkable step forward based on LSI (Hoffman 1999). pLSI is essentially a generative probabilistic topic model where each word in the document is considered to be a sample from a mixture model, and the mixture topic is multinomial distribution. As shown in Figure 19.2a, for each document in the corpus, each word \( w \) is generated from the latent topic \( x \) with the probability \( p(w|d) \), where \( x \) is from the topic distribution \( p(x|d) \) of the document \( d \). pLSI models the probability of occurrence of a word in document as the following mixture of conditionally independent multinomial distributions:

\[
p(w, d) = p(d) p(w|d)
\]

where

\[
p(w|d) = \sum_x p(w|x)p(x|d)
\]

Although pLSI proves to be useful, it has an issue in that there is no probabilistic generative model at a document level to generate the proportions of different topics. LDA was proposed to fix this issue by constructing a three-level hierarchical Bayesian probabilistic topic model (Blei et al. 2003), which is a generative probabilistic model of a corpus. Probabilistic topic models are exemplified by LDA. In the following, we are going to introduce LDA in the environment of natural language processing, where we are dealing with text documents.

A topic is defined as a distribution over a fixed vocabulary. Intuitively, documents can exhibit multiple topics with different proportions. For example, a book on topic models can have several topics, such as probabilistic general models, LSI, and LDA, with LDA having the largest proportion. We assume that the topics are specified before the documents are generated. Thus, all the documents in the corpus share the same set of topics but with different distribution. In the generative probabilistic model, data arise from the generative process with hidden variables.

In LDA, documents are represented as random mixtures over latent topics, and the prior distribution of the hidden topic structure is described by Dirichlet distribution. Each word
is generated by looking up a topic in the document it refers to and finding out the probability of the word within that topic, where we treat each topic as a bag of words and apply the language model. As depicted in Figure 19.2, LDA is three-level hierarchical probabilistic model, while the pLSI has only two levels. LDA takes the advantage of defining document distribution over multiple hidden topics, compared to pLSI, where the number of topics is fixed.

The three-level LDA model is described as a probabilistic graphical model in Figure 19.2, where a corpus consists of \( N \) documents and each document has a sequence of \( D \) words. The parameter \( \theta \) describes the topic distribution over the vocabulary, assuming that the dimensionality of the topic variable \( x \) is known and fixed. The parameter \( t \) describes the topic mixture at the document level, which is multinomially distributed over all the topics: \( \text{Dir}(\beta) \). Remember that the topics are generated before the documents, so both \( \beta \) and \( \theta \) are corpus-level parameters. The posterior distribution of the hidden variables given a document, which is the key problem we need to solve LDA, is described in the following equation:

\[
p(t, x | w, \theta, \beta) = \frac{p(t, x, w | \theta, \beta)}{p(w | \theta, \beta)}
\]

where

\[
p(t, x, w | \theta, \beta) = p(t | \beta) \prod_{n=1}^{D} p(x_n | t) p(w_n | x_n, \theta)
\]

The topic structure is the hidden variable, and our goal is to discover the topic structure automatically with the observed documents. All we need is a posterior distribution, the conditional distribution of the hidden topic structure given all the documents in the corpus. A joint distribution over both the documents and the topic structure is defined to compute this posterior distribution. Unfortunately, the posterior distribution is intractable for exact inference. However, there are a wide variety of approximate inference algorithms developed for LDA, including sampling-based algorithms such as Gibbs sampling and variational algorithms. The sampling-based algorithms approximate the posterior with samples from an empirical distribution. In Gibbs sampling, a Markov chain is defined on the hidden topic variables for the corpus. The Monte Carlo method is used to run the chain for a long time and collect samples and then approximate the posterior distribution with these samples as the posterior distribution is the limiting distribution of the Markov chain. On the other hand, variational inference is a deterministic method to approximate the posterior distribution. The basic idea is to find the nearest lower band by optimization. The optimal lower band is the approximated posterior distribution.

There are many other probabilistic topic models, which are more sophisticated than LDA and usually obtained by relaxing the assumptions of LDA. A topic model relaxes the
assumption of bags of words by generating words conditionally on the previous words (Wallach 2006). A dynamic topic model relaxes the assumption by respecting the ordering of the documents (Blei and Lafferty 2006). The third assumption that the number of topics is fixed is relaxed by the Bayesian nonparametric (BNP) topic model (Teh et al. 2006). The correlated topic model (Blei and Lafferty 2007) captures the insight that topics are not independent of each other. Instead, topics might provide evidence for each other. Other more advanced topic models, for example, the author–topic model (Rosen-Zvi et al. 2004) and relational topic model (Chang and Blei 2010), consider additional information such as authors of documents and links between the documents.

19.4 COUPLINGS AMONG TOPIC MODELS, CLOUD COMPUTING, AND MULTIMEDIA ANALYSIS

19.4.1 Large-Scale Topic Modeling

Topic modeling such as LDA enables us to discover the hidden topic structure running through the data without any supervision, which is quite useful for handling today’s big data at a scale that we can hardly annotate with human power. However, the real-world applications of topic modeling are limited due to scalability issues. When the number of documents and the vocabulary size reach the million or even the billion level, which is quite common now in practical processing, we need to boost the scalability of topic modeling to very large data sets by using a distributed computing framework. The broadly applicable MapReduce framework proves to be powerful for large-scale data processing. Thus, it is highly desirable to implement parallelized topic modeling algorithms in the MapReduce programming framework.

The challenge of scaling topic modeling to very large document collections has driven lots of effort in this research area to establish distributed or parallelized topic models. A distributed learning algorithm for LDA was proposed to improve the scalability by modifying the existing Gibbs sampling inference approach (Newman et al. 2007). The learning process is distributed on multiple processors, and each processor performs a local Gibbs sampling iteration. This approximate distributed LDA (AD-LDA) successfully opens the door for parallel implementation of topic models for large-scale computing. PLDA is based on AD-LDA, implemented on MPI and MapReduce (Chang et al. 2009). Similarly, PLDA distributes all input documents over multiple processors. Each processor maintains a copy of word-topic counts, recomputes the word-topic counts after each local Gibbs sampling iteration, and then broadcasts the new word-topic count to all other processors via communication. In the MapReduce framework, each Gibbs sampling iteration is modeled as a map–reduce task. In the Map step, map workers do the Gibbs sampling task, while in the Reduce step, reduce workers update the topic assignments. All workers, both in the Map step and in the Reduce step, work in parallel, and the master processor constantly checks the worker status to manage failure at worker processors. Thus, PLDA boosted by MapReduce provides very robust and efficient data processing.

However, Gibbs sampling has its drawbacks for large-scale computing. First, the sampling algorithms can be slow for high-dimensional models. It has been demonstrated that the PLDA can yield very good performance with a small number (millions) of documents.
with a large cluster of computers. However, it might not be able to process tens to hundreds of millions of documents, while today’s Internet data, such as Yahoo and Facebook user profiles, can easily beat this level. To handle hundreds of millions of user profiles, scalable distributed inference of dynamic user interests was designed for a time-varying user model (Ahmed et al. 2011). User interests can change over time, which is very valuable for prediction of user purchase intent. Besides, the topics can also vary over time. Thus, the topic model for user behavior targeting should be dynamic and updated online. However, it is very challenging for inference of a topic model for millions of users over several months. Even for an approximate inference, for example, Gibbs sampling, it is also computationally infeasible. While sequential Monte Carlo (SMC) could be a possible solution, the SMC estimator can quickly become too heavy in long-range dependence, which makes it also infeasible to transfer and update so many computers. Instead, only forward sampling is utilized for inference so that we do not need to go back to look at old data any more. The new inference method has been demonstrated to be very efficient for analyzing Yahoo user profiles, which provides powerful tools for web applications such as online advertising targeting, content personalization, and social recommendations.

Second, the Gibbs sampler is highly tuned for LDA, which makes it very hard to extend for other applications. To address this issue, a flexible large-scale topic modeling package in MapReduce (Mr.LDA) was proposed by using the variational inference method as an alternative (Zhai et al. 2012). Compared to random Gibbs sampling, variational inference is deterministic given an initialization, which ensures the same output of each step no matter where or when the implementation is running. This uniformity is important for a MapReduce system to check the failure at a worker node and thus have greater fault tolerance, while for random Gibbs sampling, this is very hard. Thus, variational inference is more suitable for a distributed computing framework such as MapReduce.

Variational inference is meant to find variational parameters to minimize the Kullback–Leibler divergence between the variational distribution and the posterior distribution, which is the target of the LDA problem, as mentioned earlier. The variational distribution is carefully chosen in such a way that the probabilistic dependencies from the true distribution are well maintained. This independence results in natural palatalization of computing across multiple processors. Furthermore, it only takes dozens of iterations to converge for variational inference, while it might take thousands for Gibbs sampling. Most significantly, Mr.LDA is very flexible and can be easily extended to efficiently handle online updates.

While distributed LDA has proven to be efficient to solve large-scale topic modeling problem, regularized latent semantic indexing (RLSI) is another smart design for topic modeling parallelization. The text corpus is represented as a term–document matrix, which is then approximated by two matrices: a term–topic matrix and a topic–document matrix. The RLSI is meant to solve the optimal problem that minimizes a quadratic loss function on term–document occurrence with regularization. Specifically, the topics are regularized with $l_1$ norm and $l_2$ for the documents. The formulation of RLSI is carefully chosen so that the inference process can be decomposed into many subproblems. This is the key ingredient that RLSI can scale up via parallelization. With smooth regularization, the over-fitting problem is effectively solved for RLSI.
The most distinguishing advantage of RLSI is that it can scale to very large data sets without reducing vocabulary. Most techniques reduce the number of terms to improve the efficiency when the matrix becomes very large, which improves scalability at the cost of decreasing learning accuracy.

19.4.2 Topic Modeling for Multimedia

The richness of multimedia information in the form of text, graphics, images, audio, and video has driven the traditional IR algorithms for text documents to be extended to multimedia computing. While topic modeling proves to be an effective method for organizing text documents, it is highly desirable to propose probabilistic topic models for multimedia IR. Indeed, in recent years, topic models have been successfully employed to solve many computer vision problems such as image classification and annotations (Barnard et al. 2003; Blei and Jordan 2003; Bosch et al. 2006; Duygulu et al. 2002; Li and Perona 2005; Russell et al. 2006; Wang et al. 2009).

Similar to text retrieval, image files are growing exponentially so that it is infeasible to annotate images with human power. Automatic image annotation as well as automatic image region annotation is very important to efficiently process image files. Other tasks, such as content-based image retrieval and text-based image retrieval, are quite-popular research problems. The former retrieves a matched image with a query image, which can be a sketch or simple graph, while the latter finds corresponding image results by text query. Obviously, the latter is harder as we need to establish the relationship between the image and text data, while the former can be done with visual image features. These image retrieval tasks as a whole, however, are much more difficult compared to text retrieval. Words for text document retrieval can be indexed in vocabulary, while most images are based on creation, which is rather abstract, and it is very hard to teach a machine to recognize the theme of an image (Datta et al. 2005). In the past decade, a lot of research efforts have been done in this area to make computers organize and annotate images.

Correspondence LDA (Corr-LDA) (Blei and Jordan 2003) models annotate data where one type of data is a description of another; for example, a caption and an image. The model establishes the conditional relationship between the image regions considered as latent variables and sets of words. In the generative process, the image is first generated, which consists of regions, like the topics for text documents. Then each caption word is generated from a selected region. Thus, the words of captions must be conditional on factors presented in the image. The correspondence, however, need not be one to one. Multiple caption words can come from the same region. With typical variational inference, the Corr-LDA model is demonstrated to provide much better performance for the tasks of automatic annotation and text-based image retrieval compared to the ordinary Gaussian mixture model and Gaussian-multinomial LDA.

So far, the dominant topic model LDA has been demonstrated to be very powerful in discovering hidden topic structure as lower-dimensional latent representation for originally high-dimensional features, and Corr-LDA is very effective in fulfilling tasks in multimedia retrieval. Essentially, a topic vector is a random point in a topic simplex parameterized by a multinomial distribution, and topic mixing is established by drawing each word repeatedly.
from the topics of this multinomial. While this Bayesian network framework does make sampling rather easy, the conditional dependencies between hidden variables become the bottleneck of efficient learning for the LDA models. A multiwing harmonium (MWH) was proposed for captioned images by combining a multivariate Poisson distribution for word count in a caption and a multivariate Gaussian for a color histogram of the image (Xiang et al. 2005). Unlike all the aforementioned models, MWH can be considered as an undirected topic model because of the bidirectional relationship between the latent variables and the inputs: The hidden topic structure can be viewed as predictors from a discriminative model taking the inputs, while it also describes how the input is generated. This formalism enjoys a significant advantage for fast inference as it maintains the conditional independence between the hidden variables. MWH has multiple harmoniums, which group observations from all sources using a shared array of hidden variables to model the theme of all sources. This is consistent with the fact in real applications that the input data unnecessarily come from a single source. In the paper, a contrastive divergence and variational learning algorithms are both designed and evaluated on a dual-wing harmonium (DWH) model for the tasks of image annotation and retrieval on news video collections. DWH is demonstrated to be robust and efficient with both learning algorithms. It is worth noting that while the conditional independence enables fast learning for MWH, it also makes learning more difficult. This trade-off can be acceptable for off-line learning, but in other cases, it might need further investigation.

While image classification is an independent problem with image annotations, the two tasks can be connected as they provide evidence for each other and share the goal of automatically organizing images. A single coherent topic model was proposed for simultaneous image classification and annotation (Wang et al. 2009). The typical supervised topic modeling sLDA (Blei and McAuliffe 2007) for image classification was extended to multiclass and embedded with a probabilistic model of image annotation followed by substantial development of inference algorithms based on a variational method. The coherent model, examined on real-world image data sets, provides comparable annotation performance and better than state-of-the-art classification performance.

All these probabilistic topic models estimate the joint distribution of caption words and regional image features to discover the statistical relationship between visual image features and words in an unsupervised way. While the results from these models are encouraging, simpler topic models such as latent semantic analysis (LSA) and direct image matching can also achieve good performance for automatic image annotation through supervised learning where image classes associated with a set of words are predefined. A comparison study between LSA and probabilistic LSA was presented with application on the Coral database (Monay and Gatica-Perez 2003). Surprisingly, the result has shown that simple LSA based on annotation by propagation outperformed probabilistic LSA on this application.

19.4.3 Large-Scale Computing in Multimedia

The amount of photos and videos on popular websites such as Facebook, Twitter, and YouTube is now on the scale of tens of billions. This real-life scenario is two folded. For one thing, with such huge multimedia data sets, it is even much more necessary for strategies
such as topic models to automatically organize the multimedia big data by capturing the intrinsic topic structure, which makes good sense in simplifying the big data analysis. For another, it is prohibitively challenging to handle such large-scale multimedia content as images, audio, and videos. Cloud computing provides a new generation of computing infrastructure to manage and process big data by distribution and parallelization. Multimedia cloud computing has become an emerging technology for providing multimedia services and applications.

In cloud-based multimedia computing, the multimedia application data can be stored and accessed in a distributed manner, unlike traditional multimedia processing, which is done on client or server ends. One of the big challenges with multimedia data is associated with the heterogeneous data types and services such as video conferencing and photo sharing. Another challenge is how to make the multimedia cloud provide distributed parallel processing services (Zhu et al. 2011). Correspondingly, parallel algorithms such as parallel spectral clustering (Chang et al. 2008), parallel support vector machines (SVMs) (Chang et al. 2007), and the aforementioned PLDA provide important tools for mining large-scale rich-media data with cloud service. More recently, several large-scale multimedia data storage and processing methods were proposed based on the Hadoop platform (Lai et al. 2013; Kim et al. 2014). PyCASP, a Python-based framework, is also presented for multimedia content analysis by automatically mapping computation onto parallel platforms from Python application code to a variety of parallel platforms (Gonina et al. 2014).

### 19.5 LARGE-SCALE TOPIC MODELING FOR MULTIMEDIA RETRIEVAL AND ANALYSIS

As mentioned in Section 19.4.3, it is highly desirable to develop efficient algorithms for multimedia retrieval and analysis at a very large scale, and it is quite a natural idea to develop topic models given the success of large-scale topic modeling and topic modeling for multimedia analysis, which is demonstrated in Sections 19.4.1 and 19.4.2. However, it is not an easy task to build effective topic models and develop efficient inference algorithms for large-scale multimedia analysis, due to the exponential growth of computing complexity.

The web image collection is perhaps the largest photo source. The current web image search service provided by Google or Yahoo mostly employs text-based image retrieval without using any actual image content, that is, only analyzing images based on the surrounding text features. This is very applicable as the HTML code of web pages contains rich context information for embedded images annotation. There is, correspondingly, another class of algorithms, named content-based image retrieval, which tries to retrieve images based on analysis of image regional content. Instead of purely considering the surrounding text or the visual features of the image content, such as the color and texture, it is increasingly important to develop a semantic concept model for semantic-based applications such as images and video retrieval as well as effective and efficient management of a massive amount of multimedia data.

When the amount of data scales to the tens of billions, the computational cost for feature extraction and machine learning can be disastrous. For example, some kernel machines, such as SVMs, have computing complexity at least quadratic to the amount of data, which
makes it infeasible to track ever-growing data at the level of terabytes or petabytes. To solve
the scalability issues, data-intensive scalable computing (DISC) was specially designed as a new computing diagram for large-scale data analysis (Bryant 2007). In this computing diagram, data itself is more emphasized, and it takes special care to consider constantly growing and changing data collections besides performing large-scale computations. The aforementioned MapReduce framework is a popular application built on top of this diagram. The intuition for MapReduce is very simple: If we have a very large set of tasks, we can easily tackle the problem by hiring more workers, distributing the tasks to these workers, and finally, grouping or combining all the results from the parallelized tasks. MapReduce is simple to understand; however, the key problem is how to build semantic concept modeling and develop efficient learning algorithms that can be scalable to the distributed platforms.

In recent years, the implementation of topic modeling of multimedia data on the MapReduce framework has provided a good solution for large-scale multimedia retrieval and analysis. An overview of MapReduce for multimedia data mining was presented for its application in web-scale computer vision (White et al. 2010). An efficient semantic concept named robust subspace bagging (RSB) was proposed, combining random subspace bagging and forward model selection (Yan et al. 2009). Consider the common problem of over-fitting due to the high dimensionality of multimedia data; this semantic concept model has remarkably reduced the risk of over-fitting by iteratively searching for the best models in the forward model selection step based on validation collection. MapReduce implementation was also presented in the paper and tested on standard image and video data sets. Normally, tasks in the same category of either a Map task or a Reduce task would be assumed to require roughly the same execution time so that longer tasks would not slow down the distributed computing process. However, the features of multimedia data are heterogeneous with various dimensions; thus, the tasks cannot be guaranteed to have similar execution times. Therefore, extra effort should be made to organize these unbalanced tasks. A task scheduling algorithm was specially designed in the paper to estimate the running time of each task and optimize task placement for heterogeneous tasks, which achieved significantly improved performance compared to the baseline SVM scheduler.

Most of the multimedia data we talked about are text and images. Actually, large-scale topic modeling also finds success in video retrieval and analysis. Scene understanding is an established field in computer vision for automatic video surveillance. The two challenges are the robustness of the features and the computing complexity for scalability to massive data sets, especially for real-world data. The real-world data streams are unbounded, and the number of motion patterns is unknown in advance. This makes traditional LDA inapplicable as the number of topics of LDA is usually fixed before the documents are generated. Instead, BNP models are more appealing to discover unknown patterns (Te and Jordan 2010). The selection of features and the incremental inference corresponding to a continuous stream are crucial to enable the scalability of the models for large-scale multimedia data.

A BNP model was designed for large-scale statistical modeling of motion patterns (Rana and Venkatesh 2012). In the data preprocessing stage, coarse-scale low-level features are
selected as a trade-off between model and complexity; that is, a sophisticated model can
define a finer pattern but sacrifice computational complexity. A robust and efficient prin-
ciple component analysis (PCA) was utilized to extract sparse foregrounds in video. To
avoid the costly singular vector decomposition operation in each iteration, rank 1 con-
straint due to the almost-unchanged background during a short period was used for
PCA. Then the Dirichlet process mixture was employed to model the sparse components
extracted from PCA. Similar to the generative model for text documents, in the bag-of-
event model, each feature vector was considered as a sample from motion pattern. The
combination of multinomial distribution and mixture of Gaussians defines the location
and quantity of the motion. A decayed MCMC incremental inference was developed for
fixed-cost update in the online setting. The posterior can be guaranteed to converge to the
true value on the condition that any point in past would be selected at nonzero probability.
A traditional decay function such as exponential decay only depends on time, while for
the scene-understanding problem, the probability distribution also depends on data in the
clustering space, which is the distance between the past observations and the current ones.
To improve the scalability to large-scale settings, the distance between clusters is measured
to represent the distance between samples in each cluster. The framework was tested on a
140-h-long video, whose size is quite large, and it was demonstrated to provide comparable
pattern-discovering performance with existing scene-understanding algorithms. This
work enhances our ability to effectively and efficiently discover unknown patterns from
unbounded video streams, providing a very promising framework for large-scale topic
modeling of multimedia data.

19.6 CONCLUSIONS AND FUTURE DIRECTIONS

The Internet and social media platforms such as Facebook, Twitter, and YouTube have been
so prevalent that in every, minute millions and billions of new media data such as photos
and videos are generated. Users are overwhelmed by so much information, and it becomes
rather difficult and time consuming to digest a vast amount of information. Probabilistic
topic models from the IR field are highly desirable to effectively organize large volumes of
multimedia data by discovering the hidden topic structure. On the other hand, traditional
topic models such as pLSI and LDA are hard to scale to massive data sets at the level of tera-
bytes or petabytes due to the prohibitive computational complexity, which is also the curse
for many traditional machine learning and data mining algorithms. With the emergence
of cloud computing, topic models are expected to become increasingly applicable to multi-
media data. Google’s MapReduce and Yahoo’s Hadoop are notable examples of large-scale
data processing. These techniques have demonstrated great effectiveness and efficiency for
big data analysis tasks.

While the success of topic models has been well established for text documents, much
effort has been recently devoted to their application to large-scale multimedia analysis,
where the multimedia data can be in the form of text, images, audio, and video. It is evident
from the previous sections that topic modeling is a very useful and powerful technique
to address multimedia cloud computing at a very large scale (Barnard et al. 2003; Blei and
Jordan 2003; Bosch et al. 2006; Duygulu et al. 2002; Li and Perona 2005; Russell et al.
For example, comparative studies between an ordinary Gaussian mixture model and Corr-LDA have demonstrated that Corr-LDA is significantly better for the tasks of image automatic annotation and text-based image retrieval. Various distributed and parallel computing frameworks such as PLDA and PSVM (Chang et al. 2007) have been proposed to solve scalability issues with multimedia computing. The DISC diagram is another popular design for large-scale data analysis (Bryant 2007). The popular big data processing framework MapReduce also finds its success in multimedia cloud computing applications such as web-scale computer vision (White et al. 2010). Another major issue with high dimensionality of large-scale data is over-fitting. A novel semantic concept model was proposed (Yan et al. 2009) to address the over-fitting issue, and the results have demonstrated the effectiveness of the method.

In future work, scalability will be a more prominent issue with the ever-growing data size. Web-based multimedia data mining (White et al. 2010) demonstrates the power to implement computer vision algorithms such as training, clustering, and background subtraction in the MapReduce framework. Cloud-based computing will remain a very active research area for large-scale multimedia analysis. Moreover, for cloud-based large-scale multimedia computing, it is very desirable to visualize high-dimensional data with easy interpretation at the user interface. The current method mainly displays topics with term frequency (Blei 2011), which is limited as hidden topic structure also connects different documents. The relations among multimedia data and the underlying topics are even more complex. To visualize these complex relations will significantly help people consume multimedia content. In addition, topic models are usually trained in an off-line fashion, where the whole batch of data is used once to construct the model. This process does not suit the online setting, where data streams continuously arrive with time. With online learning (Barbara and Domeniconi 2008), we do not need to rebuild the whole topic model when new data arrive but just incrementally update the parameters based on the new data. Thus, online learning could provide a much more efficient solution to multimedia analysis as some multimedia content is streaming in nature. Furthermore, feature selection and feature engineering are crucial for multimedia analysis and retrieval. With the recent advances in Deep Learning (Hinton et al. 2006; Hinton and Salakhutdinov 2006), learning a compact representation and features of multimedia data will become an important research topic.

REFERENCES


